(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
7 February 2002 (07.02.2002)

PCT

(10) International Publication Number
WO 02/10985 A2

(51) International Patent Classification⁷: G06F 17/30

(21) International Application Number: PCT/US01/23146

(22) International Filing Date: 23 July 2001 (23.07.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
0018645.2 28 July 2000 (28.07.2000) GB

(71) Applicant *(for all designated States except US)*: **TENARA LIMITED** [GB/US]; 275 Third Street, Cambridge, MA 02142 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **HADJIYIANNIS, Geroge, Ioannou** [CY/US]; 636 Beacon Street, #305, Boston, MA 02215 (US). **MUI, Lik** [US/US]; 305 Memorial Drive, Cambridge, MA 01239 (US). **ZELEVINSKY, Vladimir** [US/US]; 19 Winchester Street, #701, Brookline, MA 02446 (US).

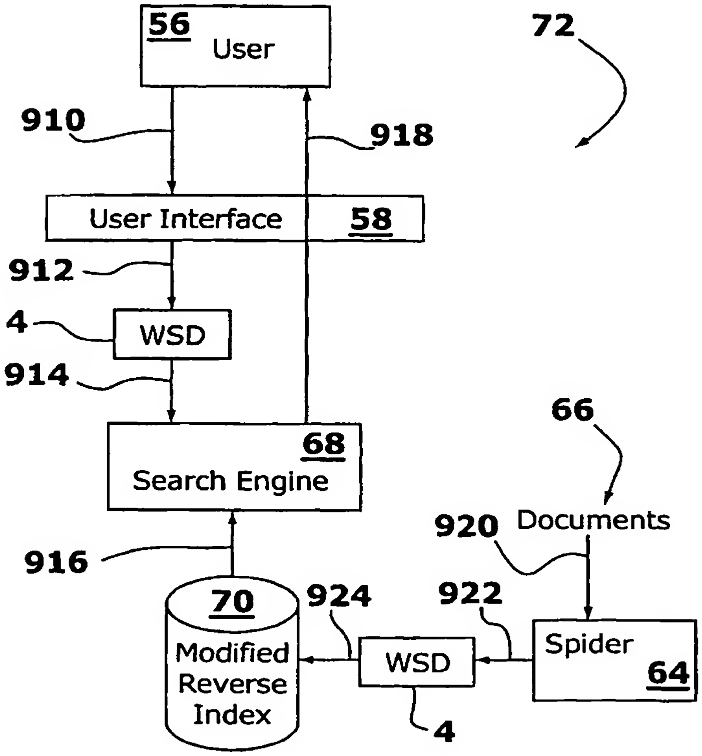
(74) Agent: **HAMILTON, John, A.**; Choate, Hall & Stewart, Exchange Place, 53 State Street, Boston, MA 02109 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian

[Continued on next page]

(54) Title: METHOD OF AND SYSTEM FOR AUTOMATIC DOCUMENT RETRIEVAL, CATEGORIZATION AND PROCESSING



(57) Abstract: A system and method are presented for performing word sense disambiguation using semantic networks with a mathematical formalism including probabilities and equivalent metrics. Also presented are techniques for automatically creating a knowledge base (network) to be used in the WSD process. Enhanced retrieval and categorization systems are developed. Information entropy theory is applied to determine importance o senses identified. Numerous applications of these basic techniques in improving accuracy of existing systems are described.

patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report*

METHOD OF AND SYSTEM FOR AUTOMATIC DOCUMENT RETRIEVAL, CATEGORIZATION AND PROCESSING

Field of the Invention

The present invention relates generally to the application of mathematical formalism to word sense disambiguation. More specifically, the invention employs probabilities and equivalent metrics to word sense disambiguation techniques in methods for document retrieval, categorization and processing. With the use of an automatically created knowledge base, systems and methods described herein have numerous applications in any field requiring knowledge management.

Background of the Invention

The information age, and the World Wide Web in particular, have caused massive amounts of information to be made available to individuals and organizations. Most of this information is in the form of electronic documents in various forms (e-mail, presentations, reports, spreadsheets, etc.) and various formats (Word, Excel, PowerPoint, PDF, PostScript, ASCII, etc.). Unfortunately, the sheer volume of information available makes it hard to locate and process information by manual means as it involves massive amounts of documents (a phenomenon known as "Information Overload" or "Digital Overload"). Automated means of locating (retrieving), organizing (categorizing) and processing such documents are necessary in order for this information to become useful to the individuals and organizations that wish to deal with such information. *See* C. Waltner, *Antidote For Information Overload:Online software lets smaller businesses cope with floods of information*, Information Week, October 2, 2000; P. Maes, *Agents that reduce work and information overload,* Communications of the ACM, 37(7):30-40, July 1994.

The two main tasks are retrieval and categorization. Retrieval is defined as locating and presenting to a user all documents pertaining to a certain topic of interest to that user. Categorization is defined as identifying a main topic or topics discussed in a document of interest.

Combining these two tasks, either with each other or with other simple, existing techniques, allows one to perform many other tasks in document processing. For example, given a task of locating and presenting documents that contain background information on a topic dealt with in an article (called a sample document), one could first perform categorization on the sample document to determine a main topic of interest, and then use this topic to retrieve additional information on this topic. Or, in another example if one wished to locate a human expert to contact on a particular topic, one could perform retrieval on the topic of interest and identify the authors of all returned documents, rank the authors in accordance with various metrics (number of documents authored, reputation, number of references to author's documents, etc.), and return a list of experts.

Abstract concepts, which are often used to represent topics, are called *senses*. Words are often called *terms*.

The performance of various retrieval and categorization mechanisms is measured by two parameters, *precision* and *recall*.

Precision is a metric that represents the accuracy of a particular approach. With respect to retrieval, precision is defined as the number of documents returned that are truly related to the topic of interest expressed as a percentage of the total number of documents returned. In terms of categorization, precision is defined as the number of topics returned that are truly part of the document expressed as a percentage on the total number of topics returned.

Recall is a metric that represents how comprehensive a particular approach is. With respect to retrieval, recall is defined as the number of relevant (to the particular topic) documents returned expressed as a percentage of the total number of relevant (to the particular topic) documents present in the system. In terms of categorization, recall is defined as the number of topics returned expressed as a percentage of all topics present in the document.

A number of techniques exist for automatically performing retrieval and categorization as well as other processing of documents:

- Indexing and Reverse Indexing (also known as keyword searching): In this approach, a topic is expressed as a combination of terms. Documents to be returned are the documents that contain all such terms. Indexing and reverse indexing techniques are only useful for retrieval. Simple indexing and reverse indexing techniques have poor precision and recall.

- Probabilistic Language Models: In this approach, documents are divided into categories, each category being relevant to a particular topic. Given a sample of documents from each category, the system calculates, for each term, the probability that it will appear in a document of the appropriate category, and that it will not appear in any documents not in the category. Thus, on arrival of a new document, the probabilities associated with each term can be used to determine the probability that the new document belongs to a particular category or not according to how often each term appears in this new document. Probabilistic Language Models are well suited to categorization. They can also be used to perform retrieval by reversing the process of categorization. Given a topic of interest (effectively a category), the terms that make it most likely that a document belongs to this category can be used as keywords to retrieve relevant documents. Probabilistic language models effectively determine a context for the document at hand (the category or categories it belongs to) and can thus achieve higher precision and recall than simple keyword methods.

- Collaborative Filtering: In this approach, the preferences of each user of a system are used to collaboratively determine the right answer. For retrieval, once a user types a query and views the returned results, he or she is asked to rate the documents according to relevance. This information is gathered from each user that types such a query and used to refine the answers presented to subsequent users typing similar queries. In effect, the system allows users to collaborate by distributing the experience of each user to all subsequent queries of the same topic. A similar approach is undertaken for categorization, except the answer is a list of topics rather than a list of documents. Collaborative filtering systems adapt very slowly to changes in user preferences, new documents, and changes in topic, and also suffer from very low resolution (i.e. cannot distinguish between closely related topics).

- Natural Language Processing: In this approach, special parsing techniques are used to allow the system to understand queries typed as standard questions rather than as combinations of terms. While this technique is used primarily to allow users not familiar with computer systems to use retrieval systems, it does provide additional information by decoding the structure of the query (determining parts of speech etc.), thus providing a small amount of context. This can be even more useful for categorization tasks where natural language parsing of the content of a document can provide context to complement the information of which terms are present in the document. Natural language parsing only provides a small amount of contextual information and as a result it has not been highly successful in enhancing precision and recall of existing systems.

- Rule-based systems: In this approach, sets of manually created rules are provided which consist of predicates (conditions that need to be satisfied) and actions. In a document-processing context, rule predicates are usually conditions that are satisfied when specific words are present in a document and actions are usually various forms of categorization (such as sending a particular document to a certain technical support individual based on whether or not it contains certain keywords). Rule-based systems cannot really be used for extensive retrieval tasks since each query would require an additional rule. They also suffer from low accuracy and recall when used for categorization. Additionally the rules must be created and maintained manually so such systems are fairly labor-intensive.

- Neural Network Models: In this approach, a neural network (a technique for allowing machines to learn from a set of training examples) can be used to perform categorization by being given a set of training documents and being told which categories they belong to. Because of this mode of operation, they behave almost identically with Probabilistic Language Models except for the fact that they learn much more slowly and require a much larger number of training examples.

The basic problem faced by all of the above approaches is the fact that topics usually consist of *senses* (or the abstract concepts in a user's mind) while documents only contain *terms* (or words). *See* C. D. Manning, H. Schutze, *Foundations of Statistical Natural Language Processing*, MIT Press, 1999. The relationship between senses and terms is not straightforward. In particular, a single sense may be represented by multiple terms and similarly a single term may represent more than one sense. For example, the sense (or concept) of "movie" is obviously represented by the term "movie" as well as by the term "film" and the slang term of "flick" and so on. Terms that represent the same sense are called synonyms. In this example, the terms "movie", "film", and "flick" are all synonyms of each other. However, the term "film" may refer to the concept (or sense) of "movie", of additionally to photographic films, chemical films, the verb "to film" which means to physically record a scene for the purposes of creating a movie, and so on. Concepts (or senses) that are represented by the same term are called homonyms.

The problem with homonyms and synonyms is that they can confuse automated tools. Synonyms result in lower recall since systems that look for only one term may miss documents that contain its synonyms.

Example: The topic of interest is Japanese movies. The terms used to express the topic in a keyword search would be "Japanese" and "movie". In a standard keyword search, this would return all documents containing the terms "Japanese" and "movie" but none that contain the terms "Japanese" and "film" (to the exclusion of the term "movie") even though such documents would be relevant.

Homonyms result in lower precision since documents that contain a different sense of a particular term (than the topic of interest) may be returned anyway.

> Example: The topic of interest is Japanese movies (as above). The terms used in the query are instead "Japanese" and "film". A standard keyword search will return a large number of documents relating to the Japanese makers of photographic films (e.g. Fuji) even though these documents are irrelevant to the topic.

The synonym problem can be solved relatively easily by looking up each sense in a table (dictionary or thesaurus), and making each synonym be part of the query as well. The homonym problem is much harder since just by inspecting a single term it is impossible to determine the appropriate sense. Instead, the context in which the term appears needs to be considered. This process is called *Word Sense Disambiguation* or *WSD*.

Of the above methods:

- Indexing and reverse indexing, as well as rule-based systems do not attempt to perform any WSD.

- Probabilistic Language Models and Neural Networks perform only indirect WSD. Such systems can derive the context in which terms appear and this helps limit the possible senses for each term. However, neither approach directly attempts to perform WSD.

- Collaborative filtering attempts to bypass the homonym problem by assuming that WSD was already performed by the users that are providing feedback, and that the results are evident in the feedback they provide. However, the feedback usually contains insufficient information to determine the context accurately.

- Natural Language parsing can perform a limited version of WSD by extracting the syntax portion of the context. For example, it can distinguish between the verb "play" and the noun "play". However, this is limited in applicability; there are still multiple meanings to both the verb and the noun.

While there are existing techniques for performing WSD, these have traditionally not been used in retrieval and categorization. Existing techniques can be divided into two broad categories:

1. Probabilistic (pattern matching) techniques: These are techniques that are based on machine learning. Machine learning algorithms (such as Latent Semantic Indexing, Clustering techniques, Neural Networks, etc.) are used to teach the machine to detect patterns that may be useful in determining context. For example, such techniques could eventually detect the pattern that the word "processor" (meaning "Central Processing Unit" or CPU) tends to often appear together with the word "computer" (meaning the electronic version, as opposed to the profession of calculating by hand ballistic trajectories for artillery shells), as well as the term "performance" (meaning "speed of execution" for a particular program). It could then assume that the term "processor" has two distinct meanings and learn to expect the term "processor" when it detects the terms "computer" and "performance" in the same document.

2. Semantic techniques: These are techniques that make use of semantic information (i.e. information related to the meaning of individual concepts and the relationships between the meanings of such concepts). This information is part of a knowledge base, which is pre-constructed (typically by lexicographers) and stored in a database in the computer. For example, such a technique would be aware of the various possible meanings of the word "bark" and also the fact that the verb "bark" is semantically related to the noun "dog", while the noun "bark" is semantically related to the noun "tree"). When faced

with the problem of determining whether a particular appearance of the word "bark" refers to the verb or the noun, they would look for the terms for concepts that are semantically related to it (i.e. the terms "dog" and "tree"). If they locate the term dog in close proximity they assume that the term "bark" referred to the verb (and by analogy can also figure out the meaning behind the term "dog"). If they locate the term "tree" they assume that the word "bark" referred to the noun (and by analogy can figure out the meaning of the word "tree").

Probabilistic techniques suffer from three problems:

1. Since they have no information when they start learning they need to be guided through the process. This usually involves the use of training examples (from which the machines can detect the patterns). Such training examples must be manually selected which results in substantial manual effort in the best cases, and incomplete coverage in the worst cases.

2. The patterns that are learned by such techniques are dependent on language, particular dialects and local slang. In addition, they are also sensitive to domain (i.e. the particular area of interest that the document deals with, such as financial information vs. pharmaceutical information). Therefore, probabilistic techniques need re-training when there is a change in language or domain, which involves selecting new training examples and therefore manual effort. By comparison, the information used by semantic techniques does not change with any of the above factors: the verb "bark" is semantically related to the noun "dog" no matter which language is used or which domain is applicable.

3. Since they have no notion of the distinction between different meanings of the same word, they often cannot resolve (i.e. distinguish between) closely related meanings. For example, they may be unable to distinguish between the verb

"play" (meaning to have a role in a play) and the noun "play" since they both tend to appear in the same patterns.

Because of the above problems, semantic techniques are generally better for most applications. Unfortunately, semantic techniques have two main problems of their own:

1. They assume that the semantic information is already available. The individual concepts behind inside the knowledge base must be entered manually. There are extensive lists of such concepts created by lexicographers for the purposes of creating dictionaries and thesauruses that can be used to create such a knowledge base. However, the number of semantic relationships between such concepts is much larger and such relationships are not commonly available. Manual methods of creating such relationships have been used but these are very laborious, with extensive knowledge bases taking 15 years to develop and being incomplete nonetheless. *See* D. B. Lenat. *CYC: A large-scale investment in knowledge infrastructure,* Communications of the ACM, 38(11), 1995; G.A. Miller, *WordNet: A lexical database for English,* Communications of the ACM, 38(11), pp. 39-41, 1995.

2. They treat all semantic relationships equally. Since most of these relationships are created manually, it is impossible to determine an accurate "strength" for each particular relationship. For example, there is a much stronger relationship between "car" and "engine" than between "car" and "seat" but it is impossible to quantify the strength manually.

Summary of the Invention

The present invention is directed to a system for and method of performing word sense disambiguation on a document. In accordance with the invention, a network is provided comprised of nodes corresponding to senses, edges corresponding to semantic relationships between the senses, and the probabilities of the nodes and the edges. The method is comprised of receiving a document, converting the document into a list of terms, applying a stemming algorithm to the list of terms, looking up in the network each resulting stem to determine all senses possibly referring to each stem, applying a heuristic to select likely interpretations for each set of senses, calculating the probability of each interpretation being the correct interpretation, and returning the most likely interpretation of the document. The heuristic employed may be an exact or an approximate heuristic.

In another aspect, the present invention is directed to a system for and method of performing word sense disambiguation on a document using numerical metrics. In accordance with the invention, this system and method are similar to that described above, except that in calculating the likelihood of interpretations, this embodiment defines node weights and edge weights in terms of logarithms of node and edge probabilities, defining sense weights in terms of corresponding node weights and edge weights, and summing the sense weights corresponding to each interpretation. In this embodiment, the most likely interpretation of the document is that for which the sense weight sum is a minimum. In either of the above embodiments, irrelevant edges may optionally be removed from the network.

In another aspect, the present invention is directed to a system for and method for automatically creating a network comprised of nodes, edges, and their respective probabilities. This involves converting a set of input files, such as lexicographer files, containing known senses for terms and known semantic relationships between the senses into a preliminary network, initializing count values to the nodes and edges, assigning probabilities to the nodes and edges by determining the relative frequency of each term compared to its homonyms, and the relative frequency of a particular pair of nodes corresponding to an edge compared to all other edges between nodes corresponding to the same terms as the particular pair of nodes, automatically creating additional edges identified by correlation analysis, and word sense disambiguating a large corpus of randomly selected documents to adjust the probabilities of the nodes and edges, thereby obtaining a finalized network. The additional edges may alternatively be created randomly or everywhere it is possible to create an edge in the network.

In another aspect, the present invention is directed to a system for and method of retrieval using word sense disambiguation. In accordance with the invention, the system receives a query, disambiguates the query to identify the senses corresponding to the query, entering the senses into a search engine, and retrieving via the search engine entries from a sense-based reverse index. The system may additionally be configured with a sense-based reverse index comprised of entries corresponding to senses resulting from word sense disambiguating documents collected by a crawler or a spider. In an alternative embodiment, the invention provides a query reformulation module for receiving a query, identifying the senses of the query through word sense disambiguation, and reformulating the query to include the identified senses while removing ambiguity.

The reformulated query may then be entered into a search engine to retreive entries from a reverse index. Documents corresponding to the retrieved entries may optionally be ranked by their relative importance to the query's identified senses.

In another aspect, the present invention is directed to a system for and method of categorization using word sense disambiguation. In accordance with the invention, the system provides a network comprised of nodes corresponding to senses, and edges corresponding to semantic relationships between the senses, receives a document, semantically disambiguates the document, determines the relative importance of each node by calculating the entropy of each resuting node, and applies grouping rules to the resulting set of node entropies to determine into which topical group or groups the document should be inserted. The grouping rules may optionally be created automatically by performing probablistic analysis on the correlation between the set of node entropies and the topical group or groups.

The categorization system and method described above may be employed in providing a retrieval system configured with a sense-based reverse index comprised of entries corresponding to only the most important senses identified per the categorization method and resulting from word sense disambiguating documents collected by a crawler or a spider.

In another aspect, the present invention provides a system for and method of propagating virtual edges for use in word sense disambiguating a document. The system locates intervening terms in the network between terms corresponding to terms in the document for which no directly connected semantically related pair exists in the network, then defines virtual edges between the nodes corresponding to the indirectly related document terms, wherein the virtual edge probabilities are equal to the product of the probabilities of all semantic edges connecting the nodes, disambiguates the document as described above including the intervening terms, and removes the nodes corresponding to the intervening terms.

Numerous applications of the retrieval, categorization and processing techniques described above are described in various embodiments herein. These include systems for and methods of automated document translation, natural language processing, grammar and syntax checking, dynamic personalization, customer relationship management, document hyper-linking, summarization of documents, speech recognition, and optical character recognition.

Brief Description of the Drawings

Figure 1 is a block diagram of an embodiment of computer system configured with a WSD module.

Figure 2 is a schematic diagram depicting a simple network for WSD.

Figure 3 is a schematic diagram depicting an extended network for WSD.

Figure 4 is a flow diagram illustrating an embodiment of the WSD process in accordance with the invention.

Figure 5 is a flow diagram illustrating an embodiment of the selection heuristic (modified gradient descent) process employed in interpretation selection in accordance with the invention.

Figure 6 is a schematic diagram illustrating a network with counts assigned to nodes and edges.

Figure 7 is a flow diagram illustrating the network creation process in accordance with the invention.

Figure 8 is a block diagram illustrating a standard search engine structure.

Figure 9 is a block diagram illustrating a modified search engine structure for retrieval using WSD.

Figure 10 is a block diagram illustrating a search engine structure employing a query reformulation module.

Figure 11 is a flow diagram illustrating the categorization process using WSD in accordance with the invention.

Figure 12 is a block diagram illustrating a modified search enginer structure to improve retrieval using WSD categorization.

Figure 13 is a schematic diagram illustrating a network including a sample virtual edge.

Figure 14 is a flow diagram illustrating the edge propagation process in accordance with the invention.

Detailed Description

Preferred embodiments of the invention will now be described with reference to the accompanying drawings.

In one aspect, the present invention is directed to systems for and methods of enhancing the performance of each of retrieval and categorization using WSD techniques. By performing WSD on a query, a system can automatically determine the meanings behind each keyword contained in the query. The method can be used to solve both the synonym and homonym problems identified above. Similarly, once the meaning of each word in a document has been determined, techniques based on information entropy theory may be used to determine how important each word and its associated meaning is to the document. These techniques, described below, are useful in performing categorization.

In another aspect, the present invention is directed to systems for and methods of performing WSD using semantic information, but including a mathematical formalism to increase accuracy for use in retrieval and categorization. Standard semantic WSD techniques operate by trying to find terms for semantically related concepts in close proximity to the term being disambiguated. However, without a mathematical formalism that can express the strength of such relationships and take into account factors of importance (such as positioning), it would be impossible to make a determination in cases where multiple meanings of a word have semantically related terms in close proximity. Using Bayesian inference rules, one can create a formalism where the most "likely" interpretation for the whole document is selected by calculating the probability that each candidate interpretation is correct. This can substantially improve accuracy.

In yet another aspect, the present invention is directed to systems for and methods of creating knowledge bases required to perform WSD that are fully automated. In order to create a WSD system that allows for a mathematical formalism, one must use a knowledge base comprised of a substantial number of relationships between senses, and in addition to that, associated prior probabilities that will be used by the mathematical formalism to determine a most likely interpretation of a document. As indicated earlier, manual attempts at generating such knowledge bases require a tremendous amount of effort and cannot generate the prior probabilities necessary. Therefore, an automatic method for creating such knowledge bases is necessary. In addition, such automatic methods of generating the knowledge bases allow for re-training to new domains (should this become necessary) without the need for manual effort, thus making tools based on this method more flexible.

Figure 1 is a high-level block diagram of a general-purpose computer system **2** upon which a WSD module **4** preferably executes. The computer system **2** contains a central processing unit (CPU) **6**, input/output devices **8**, and a computer memory **10**. Among the input/output devices is a storage device **12**, such as a hard disk drive, and a computer-readable media drive **14**, which can be used to install software products, including the WSD module **4**, which are provided on a computer-readable medium. The memory **10** preferably includes the WSD module **4**, which further includes the knowledge base, or *network* **16** of the invention. In one embodiment, the computer system **2** is linked to other computers via the Internet. While the WSD module **4** is preferably implemented on a computer system configured as described above, those skilled in the art will recognize that it may also be implemented on computer systems having different configurations.

A. Word Sense Disambiguation Using Probabilities

WSD is the process by which WSD module **4** determines the sense that corresponds to each word in a document. The WSD method of the present invention combines semantic methods with a rigorous mathematical model based on probabilities to determine the most likely sense for each word in the document. To achieve this, the method uses a network **16** comprised of senses (called "nodes") and relationships between these senses (called "edges"). **Figure 2** depicts a simple example of such a network **16**. Nodes (**18a-18d**) exist in the network **16** for each possible meaning of each word that the WSD module is designed to recognize. A network also contains edges of the following three types:

a) Synonym edges (not shown in **Figure 2**): These edges relate nodes that are synonyms of each other (i.e. mean the same thing but are represented by different words). For example, nodes for "movie" and "film" would be linked by such an edge.

b) Homonym edges (**20** in **Figure 2**): These edges relate nodes (such as **18a,18b**) that are homonyms of each other (i.e. mean different things but are spelled exactly the same). For example, the verb "bark" and the noun "bark" (meaning the outer layer of a tree trunk) are linked by a homonym edge **20**.

c) Semantic edges (**22** in **Figure 2**): These edges relate nodes (concepts) that are somehow related in meaning. For example, the node **18c** for the noun "dog" and node **18b** for the verb "bark" are related by a semantic edge **22**. Similarly, the node **18a** for the noun "bark" and the node **18d** for the noun "trunk" (meaning the part of a tree) are related by such an edge **22**.

Given such a network, the basic approach to solving WSD problems can be described as follows:

- For each word in a document to be disambiguated, determine all possible nodes that could refer to it.
- Once these nodes are located, for each node locate all other nodes are semantically related to it.
- Look for these semantically related nodes in close proximity to the original word. Two terms are in close proximity if the number of intervening terms in the document is lower than a certain threshold, or they appear in the same sentence.
- The node corresponding to the original word is the one for which the preponderance of semantically related nodes were located in close proximity to the original word.

Example: Consider the phrase "Most dogs bark at postmen". The term under consideration is "bark". According to **Figure 2,** there are two possible nodes **(18a,18b)** representing the term "bark". The first node **18a** is the noun, which is semantically related to the term tree. An examination of the phrase reveals that the term "tree" does not appear in close proximity to the word under scrutiny. By comparison, the second node **18b** for the term "bark" is the verb, which is related to the term "dog". An examination of the phrase reveals the term "dog" in close proximity to the word under scrutiny. Therefore, node **18b** (representing the correct sense for the term bark) is selected.

The above description of the process does not include the mathematical formalism inherent in the present invention. The mathematical formalism is used to determine which node for the term under scrutiny has a *preponderance* of semantically related nodes in close proximity to the given term. In order to determine this we use a modified network such as is shown in **Figure 3.**

This portion of network **16** shows the different senses represented by the term "trunk", and how these relate to the different senses of the term "bark". The term "trunk" has 3 nodes associated with it: node **24a** represents the sense that is a synonym of the term "torso", node **24b** represents the part of a tree that is a synonym of the term "bole", and node **24c** represents the part of a car that is a synonym of the term “boot”. Similarly, the term “bark” is associated with two different nodes: node **26a** represents the verb “bark” which is semantically related to the node for “dog”, and node **26b** represents the noun “bark” which is semantically related to trees.

The network **16** depicted in **Figure 3** is different from that of **Figure 2** in two ways:

a) Semantic edges now run between all possible pairs of senses for the two terms. In other words, all possible pairs consisting of a node that is represented by the term “trunk” (nodes **24a, 24b,** and **24c**) and a node that is represented by the term “bark” (nodes **26a** and **26b**) are related by semantic edges **28**, rather than only nodes **24b** and **26b**, which are truly related. Two nodes are described as truly related if they are related by a known semantic relationship as defined below.

b) Each node (**24a, 24b, 24c, 26a,** and **26b**) and each of the edges **28** have a probability associated with them. These are shown as p_{nx} for nodes (where x is the node number) and $p_{ex,y}$ for edges (where x and y are the node numbers for the nodes related by the edge).

One assumption made is that even though edges exist between all possible pairs of senses for the two terms, the probabilities reveal which ones are truly related to each other. In other words, edges between nodes that are truly related semantically will have high probabilities associated with them and edges between nodes that are not truly related will have very low probabilities associated with them.

Let the term corresponding to node x be defined as *term(x)*. The probability p_{nx} is defined as the probability that node x is the correct sense for *term(x)* given that *term(x)* appears in the document but given no other information. In other words, if *term(x)* is found in a document and no other information is known, then p_{nx} represents the probability that node x is the correct sense for the term. Let the probability that node x is the correct sense for *term(x)* and node y is the correct sense for *term(y)*, given that *term(x)* and *term(y)* appear in the document and no other information, be defined as $p_{x,y}$. In other words, if *term(x)* and *term(y)* are found in the document and no other information is known, a node must be selected to represent *term(x)* and a node to represent *term(y)*. Then $p_{x,y}$ represents the probability that node *x* is the correct sense for *term(x)* and node *y* is the correct sense for *term(y)*. Given the above, one may define

Eqn 1: $$p_{ex,y} = p_{x,y} / (p_{nx} * p_{ny})$$

Intuitively, p_{nx} represents how much more common one sense for a particular term is compared to the others. For example, the term "car" has at least five different senses but the majority of the time it is used to refer to an automobile (the other senses including railway car, cable car and elevator cabin). Similarly, $p_{x,y}$ represents the semantic relation that allows one to determine that when one sees the term "dog" in close proximity to the term "bark", the term is probably referring to the family pet (as opposed to the verb which means to pursue) and the verb referring to the dog's vocalizations (as opposed to the noun which refers to the outer layers of a tree trunk).

Given the above definitions, one can define the probability that node *x* represents the correct sense of *term(x)* in an extended document as a weighted average of the above probabilities:

Eqn 2: $$P(x) = p_{nx} * (p_{ex,o} * p_{ex,p} * \ldots * p_{ex,q})^{1/c}$$

Where nodes *o, p* … *q* are senses in close proximity to node *x* that are semantically related to node *x*, and *c* is the number of such nodes.

A document may be defined to be a list of terms $t_1 \ldots t_n$. An *interpretation* of the document may be defined to be a list of senses $s_1 \ldots s_n$, one such sense being selected for each term out of all the possible senses for that term. Note that for each document there are a large number of possible interpretations (corresponding to all possible combination of each sense for each term).

Given the above definitions, one may define the probability that a particular interpretation is the correct one to be:

Eqn 3: $$P_{interpretation} = P(s_1) * P(s_2) * P(s_3) * \ldots * P(s_n)$$

Equation 3 allows the WSD module **4** to determine how likely to be correct each interpretation is. Various possible interpretations may then be compared in order to select the most likely (i.e. the one with the highest probability). Note that the number of interpretations can be very large and in general, for substantial document sizes it is impossible or impractical to compare all possible interpretations. Therefore, one or more approximations (heuristics) need to be used to locate promising interpretations and compare them to each other.

Also, note that each term may have various possible forms. For example, the verb "bark" could appear as "bark" or "barked" or "barking". While each such form could be included as a synonym in network **16**, this would make the number of nodes in the network unnecessarily large. Instead, one can try to convert each form of the word to the root (also known as stem) – in the example above, the last two forms would be converted to "bark". Existing stemming algorithms may be used for this step, such as Porter's Stemming Algorithm. Note that stemming may lead to multiple stems (e.g. "barking" could be a form of the verb "bark" or a noun), each of which may have multiple senses associated with it. All senses associated with each stem are considered as possibilities in constructing an interpretation.

Figure 4 is a flow diagram illustrating a preferred embodiment of the inventive WSD process **400** as described above. In **Step 402**, a document **30** is first converted from whatever electronic format it is written in (such as HTML, Microsoft Word, etc.) to a list of terms $t_1 \ldots t_n$ **32**. In **Step 404**, each of these terms **32** is passed through a stemming algorithm, which produces (for each term) a stem list **34**. In **Step 406**, each stem in each stem list is looked up in the network to determine all possible senses it could refer to and all such senses combined to yield a list of senses (nodes) **36** that could represent the corresponding term. After such a list of lists of senses **36** is generated, in **Step 408** a heuristic is used to select likely interpretations. These are then compared to each other according to the above probability metric and the most likely one selected and returned as a disambiguated document **38**.

The present invention, in another aspect, is a system comprised of a computer system **2** as shown in **Figure 1** capable of executing the WSD module **4** to practice the method described above.

HEURISTICS FOR INTERPRETATION SELECTION

As stated above, a document **30** may be defined to be a list of terms $t_1 \ldots t_n$ **32** (having been filtered if necessary). One may define *nodes(t_x)* to be the number of possible nodes (after stemming **404** and sense lookup **406** for each stem) that could be represented by term t_1. The number of possible interpretations (*I*) to the document is then

Eqn 4: $$I = \text{nodes}(t_1) * \text{nodes}(t_2) * \ldots * \text{nodes}(t_n)$$

If one assumes a conservative average value for nodes(t_x) to be 3 and the average document length (i.e. *n*) to be 1000 terms, then the average number of interpretations is 3^{1000}. Clearly, this number of possible interpretations is so large that it is beyond the capabilities of any existing computer system to explore all such possibilities (let alone explore them in a reasonable amount of time). In order to solve the problem, the solution must be approximated through heuristics, which explore only a small subset of the possible interpretations that hopefully contains the best interpretation.

A large number of embodiments applying different heuristics are envisioned as applicable and within the scope of the invention. Generally, heuristics may be divided into two broad categories:

a) **Exact Heuristics:** These heuristics will always compute the best interpretation. They try to locate the best interpretation early on and then prove that the rest of the interpretations will not perform any better than the current interpretation. However, in order to prove that the current interpretation is better than all other possible interpretations they may have to explore a large number (possibly all) of interpretations and, therefore can take a very long time to determine the correct answer. In general, though, they tend to arrive at a solution relatively quickly. Exact heuristics guarantee the accuracy of their answer (i.e. that they will locate the best interpretation), but cannot provide any guarantees about the time required to arrive at such an answer.

b) **Approximate Heuristics:** These heuristics tend to compute approximate answers (i.e. will not guarantee the best possible interpretation but will attempt to locate one that is very close to it). Approximate heuristics often have guarantees on the time it takes to produce an answer though.

Examples of exact heuristics include Branch-and-Bound techniques, alpha-beta search techniques (a version of Branch-and-Bound), various Integer Linear Programming techniques, matrix covering techniques (such as unate and binate covering), and others.

Examples of approximate heuristics include Linear Programming techniques, Genetic Algorithms, Simulated Annealing techniques, Gradient Descent algorithms and various modifications of the above.

Step 408 of selecting a document interpretation for a WSD implementation has the following characteristics:

a) The time required to produce an answer must be controlled and as short as practicable.

b) The best interpretation is often accompanied by a large number of good interpretations with probabilities that are very close to the best interpretation. This is because primary senses in the document typically have many nodes that are related to them semantically (and thus make it easy to disambiguate all such nodes) but there may be incidental words whose meaning does not change the meaning of the document much (i.e. they are not closely related to any of the other nodes in the document). The sense assigned to these terms can vary without changing the probability of the interpretation too much, but at the same time, the sense assigned to such terms is not important to the meaning of the document.

Given the characteristics listed above, in its preferred embodiments the invention employs Approximate Heuristics (although using exact heuristics is obviously not impossible or impracticable under certain circumstances). More particularly, a modified Gradient Descent algorithm is employed in interpretation selection. The steps of this modified algorithm are shown in **Figure 5** and are as follows:

Step 510: For each sense set s_j such that *i* is not equal to *j*, examine all cross-connected pairs consisting of a node from set s_i and a node from set s_j. For each such pair, compute P(*i*) (where only the senses in the pair are included in the calculation). Select the sense that results in the maximum P(*i*) as the initial sense for set s_i.

Step 520: If initial values have been selected for all the sense sets then proceed to **Step 530**, otherwise perform **Step 510** for the next sense set.

Step 530: Assuming all other senses in the current interpretation remain unchanged, for each sense in sense set s_i, compute P(*i*) (where all senses in the current interpretation are included in the calculation). Adjust the sense selected from s_i for the current interpretation to be the one that results in the maximum P(*i*).

Step 540: If the senses for all sense sets have been adjusted, then proceed to **Step 550**, otherwise repeat **Step 530** for the next sense set.

Step 550: Repeat **Step 530** and **Step 540** until no more adjustments occur or until you have repeated at least as many times as a pre-defined threshold.

B. Generating the Networks

The WSD process **400** described earlier depends on the existence of a network **16** of the type shown in **Figure 3**, which contains accurate probabilities for all nodes and edges. Such a network contains hundreds of thousands of nodes (such as **24a, 24b, 24c, 26a,** and **26b**) and millions of edges **28**. Therefore, creating such a network is a non-trivial matter.

As described earlier, the manual effort required to produce such a network is prohibitive and more often than not results in a network with an insufficient number of nodes and edges. Furthermore, manual methods cannot produce the probabilities necessary to allow the interpretation selection algorithms to work accurately. The present invention provides a system for and method of automatically generating as much of the network as possible.

The task of creating network **16** consists of three sub-tasks:

a) Creating the nodes (senses or concepts)
b) Creating the edges (relationships)
c) Assigning probabilities to the nodes and edges

Creating nodes is a task that cannot be automated well if these nodes are to correspond to true abstract concepts. Latent Semantic Indexing (LSI) and other similar techniques can be used to discover possible hidden word associations that may indicate different meanings behind a single term but it is hard to distinguish what corresponds to a true concept as opposed to simple associations between words. For example, true concepts are invariant of language, while any association discovered by a technique such as LSI will be strongly dependent on the language used.

The desired concepts and the words that correspond to them have already been identified and organized for creating dictionaries and thesauruses. Lexicographers generate lists of such concepts and the corresponding words (and associations thereof) in standardized formats. Because of the much smaller number of nodes (rather than edges), it is possible to generate a very extensive list of nodes manually as has been done for virtually every language. Computer system **2** can easily use such files to create nodes for the network **16** automatically.

There are three types of edges that need to be created. Synonym edges can be extracted from lexicographer files used to create thesauruses. Homonym edges (such as edge **20** in **Figure 2**) can be generated by simply comparing the spelling of individual terms that correspond to nodes (such as **18a** and **18b**), once the nodes are created. Creating semantic edges (such as edges **22**) is a much harder task simply because of the very large number of edges that need to be created. Furthermore, nothing similar to the lexicographer files exists that would contain a *comprehensive* list of such information. The present invention alleviates the requirement that only edges between nodes that are semantically related exist in the network by assuming that the probabilities on edges between nodes that are not semantically related will be particularly unfavorable. This means that an excess of edges could be created provided that all nodes that are semantically related have an edge between them. In a much less preferred embodiment, edges are generated blindly between all nodes. However, given the large number of nodes, this could result in billions of edges and storing such a large number of edges is impractical. Instead, it is desirable to create edges only between all pairs of senses that could represent two terms as long as at least one of the pairs is semantically related.

Consider again the example shown in **Figure 3**. It is desired to create edges **28** between nodes **24a**, **24b**, and **24c**, and nodes **26a** and **26b**, if at least one of these pairs of nodes represents a true semantic relationship. It is allowable to create such edges even if no such relationship exists (as long as the probabilities associated with these edges give no preference to a particular pair), but such edges would be a waste of resources and, by definition, would not affect the results of the WSD process **400** since they give the same preference to all pairs.

Referring to **Figure 6**, the present invention employs correlation analysis to detect pairs of terms (such as **24b-40**, **42-26b**, etc.) that may have senses that are semantically related. Terms that have such related senses will tend to appear together slightly above average and will therefore have a correlation coefficient greater than 0. Edges (such as **44** and **46**) are created automatically between all pairs of nodes corresponding to such pairs of terms and shown as **Step 708** in **Figure 7**. The process of assigning probabilities needs to be seeded with a number of true semantic relationships, as explained below. Such semantic relationships are used to create edges between all pairs of nodes belonging to the terms that correspond to the nodes that are semantically related.

The final step to creating an appropriate network **16** is assigning probabilities to the nodes and edges in the network. In order to determine these probabilities, one needs to measure: (a) the relative frequency of each sense compared to its homonyms; and (b) the relative frequency of a particular pair of nodes corresponding to an edge, compared to all other edges between nodes corresponding to the same terms as the two original nodes.

These frequencies may be determined by counting how many times each node appears in a set of documents, and how many times nodes appear together in documents.

The nodes and edges of the network **16** depicted in **Figure 6** are annotated with such counts **48, 52**. Nodes (**24a, 24b, 24c, 26a,** and **26b**) have node counts **52** of the type c_{nx} while edges **50** have edge counts **48** of the type $c_{ex,y}$ where node(x) and node(y) are the nodes related by an edge. Given these çounts, the appropriate probabilities can be derived as follows:

Eqn 5: $p_{nx} = c_{nx} / (c_{nx} + c_{ny} + \ldots + c_{nz})$

where nodes x, y … z are all the homonyms of node x (includes node x itself)

Eqn 6: $p_{ex,p} = c_{ex,p} / (c_{ex,p} + c_{ex,q} + \ldots c_{ex,r} + c_{ey,p} + c_{ey,q} + \ldots + c_{ey,r} + \ldots + c_{ez,p} + c_{ez,q} + \ldots + c_{ez,r})$

where nodes x, y…z are the homonyms of node x and nodes p, q…r are the homonyms of node p.

The counts **48,52** may be obtained by inspecting a sufficiently large corpus of documents. However, this presumes knowledge of the senses that correspond to the terms in each document. An automated methodology for performing this step is necessary. The present invention employs a bootstrap technique to assign such probabilities by making use of lists of semantic relationships that have been manually created for other purposes but are too small to be comprehensive (such as the relationships that were manually catalogued in attempts to create semantic disambiguation systems in the past). The process is executed as follows:

a) All node counts **52** are initialized to equal values. These values are selected to be large enough so that they are not too easily swayed by statistical variations in the corpus sample, but small enough to follow distinct trends in the corpus sample. All edge counts **48** are similarly initialized to equal values except for edges that correspond to known semantic relationships. Edges that correspond to known semantic relationships are given edge counts that are substantially higher than the rest of the edge counts and therefore given preference during WSD. Once all counts **48,52** for both nodes and edges are initialized, the appropriate probabilities are computed using the above equations. This process of initializing the counts and computing preliminary probabilities from them is called initialization and is reflected as **Step 706** of **Figure 7.**

b) Using the existing probabilities (which now correspond to a limited set of known semantic relationships), a document **30** is subjected to the WSD process **400** described earlier. This can be used to determine the actual senses in the document **30** for a limited number of nodes for which one can find known semantic relationships. The node counts **52** for these nodes are correspondingly upgraded. Similarly, the edge counts **48** for all such nodes that were successfully resolved are correspondingly updated. The WSD process **400** is repeated for a very large number of documents to obtain representative counts for all nodes and edges. This process is called calibration and is reflected as **Step 710** of **Figure 7.**

The effect of calibration is two-fold. It adjusts the node probabilities to values measured from a representative corpus. And it adjusts the edge probabilities to values measured from a representative corpus irrespective of whether or not these edges correspond to the initial list of known semantic relationships. The latter effect in particular implies that the automatic generation process can generate semantic relationships that were never present in the original set of known semantic relationships. There simply needs to be an existing edge before the calibration phase; the calibration phase will automatically assign the appropriate probability to the existing edge assuming a large enough corpus is used for calibration.

Consider the example network **16** shown in **Figure 6**. Assume that edge **44** corresponds to a known semantic relationship and therefore has preference after initialization, and likewise for edge **46**. Assume that the network **16** is used to disambiguate a document that contains (among others) the terms "bole", "trunk", "bark" and "tree". By virtue of edge **44,** one can disambiguate "bole" and "trunk", and similarly, by virtue of edge **46** one can disambiguate "bark" and "tree". The edge between nodes **24b** and **26b,** does not correspond to a **known** semantic relationship but does correspond to a real semantic relationship. Therefore, nodes **24b** and **26b** are more likely to appear together than any other combination between nodes **24a, 24b,** and **24c,** and nodes **26a** and **26b**. This will result in $c_{e2,5}$ being higher than the edge counts for any of the other edges, and the eventual probability associated with this edge to be higher as well. If the edge does not correspond to a true semantic relationship (and no other edge corresponds to one either) then the probabilities will eventually end up being even according to the Law of Averages, and therefore none of the edges will affect the WSD process **400**. Therefore, this process can be used to discover true semantic relationships that were not in the set of the original known semantic relationships.

Figure 7 shows the overall process **700** of automatically creating a network:

In **Step 702,** a set of lexicographer files (and various databases that contain known semantic relationships) are automatically converted into a set of commands that a computer can use to create the nodes in a network, and the edges corresponding to the known semantic relationships. This allows the input to come form a variety of sources with non-uniform formats.

In **Step 704,** the set of commands produced in **Step 702** is processed by an automated system to create a preliminary network containing these nodes and edges.

In **Step 706**, the system then processes this preliminary network according to the initialization process described above, resulting in a new network that contains preliminary probabilities for the nodes and edges.

In **Step 708**, because the new network does not contain an extensive enough set of edges, edges are automatically added according to the results of the correlation analysis process referred to above to expand the set of edges. This results in a new network that contains the final set of nodes and edges, and preliminary probabilities for them.

Finally, in **Step 710**, the system performs the calibration process described above using a large corpus of randomly selected documents to adjust the probabilities for the nodes and edges to their final values, resulting in a finalized network **16** required by the WSD process **400**.

C. Retrieval Using Word Sense Disambiguation

Figure 8 illustrates both a standard (prior art) reverse indexing retrieval system **54** and the steps to using such a system. In such a system, the following sequence of events occurs:

Step 810: A user **56** submits a set of keywords as a query via an interface **58**;

Step 812: The set of keywords is entered into a Search Engine **60**;

Step 814: The Search Engine **60** retrieves the appropriate entries from a reverse index **62**;

Step 816: The Search engine **60** locates relevant documents and formats a response to the user **56**.

Additionally, the following events occur in the background:

Step 818: A program (called a crawler or a spider **64**) collects documents **66** from various sources (including the World Wide Web);

Step 820: These documents **66** are processed to determine which terms are contained in them and the corresponding entries are added to the reverse index **62.**

FIRST RETRIEVAL EMBODIMENT

As stated above, search engine structure **54** suffers from low precision because it cannot determine which sense is associated with each keyword and each word in the document so it cannot distinguish between homonyms. **Figure 9** illustrates a modified search engine system **72** and the steps to using the system in a retrieval method that overcomes these problems using WSD modules **4**:

Step 910: The user **56** submits a set of keywords as a query via the interface **58;**

Step 912: The set of keywords is entered into a WSD module **4;**

Step 914: The WSD module **4** produces the senses that correspond to the keywords. These senses are entered into Search Engine **68;**

Step 916: The Search Engine retrieves the appropriate entries from a modified reverse index **70** (modified in that it contains entries for each sense, rather than entries for each term);

Step 918: The Search Engine **68** locates relevant documents and formats a response to the user **56.**

To maintain the modified reverse index **70,** these tasks are accomplished prior to, or concurrently with the search:

Step 920: A crawler or spider **64** collects documents **66** from various sources (including the World Wide Web);

Step 922: Each document **66** is entered into a WSD module **4**;

Step 924: The WSD module **4** produces a sense for each word in the original document **66**. These senses are processed and entered into the corresponding entries in the modified reverse index **70**.

Since the search engine **68** receives a disambiguated query (from **Step 914**) and the entries in the reverse index are themselves disambiguated, the search engine **68** is not confronted with any word ambiguity and can thus provide much better precision. Note that synonyms can be included in the disambiguated query so as to increase recall as well.

SECOND RETRIEVAL EMBODIMENT

While the previous embodiment is preferred for retrieval using WSD, it requires a completely new system to be installed. For existing retrieval systems, an add-on module (called a Query Reformulation module, or QR) that re-writes queries so that the queries result in higher precision and recall can be implemented instead. **Figure 10** illustrates such a search engine system **76** employing a QR module **74** and the steps to using such a system in a method comprised of the following the steps:

Step 100: The user **56** submits a set of keywords as a query via an interface **58**;

Step 102: The set of keywords is entered into a WSD module **4**;

Step 104: The WSD module **4** produces the senses that correspond to the keywords. These are entered into a Re-Write module **78**;

Step 106: The Re-Write module **78** produces a new query, which will address the same topics as the original query but is unambiguous and includes all synonyms for each keyword. This reformulated query is entered into the Search Engine **68**;

Step 108: The Search Engine **68** retrieves the appropriate entries from the reverse index **62**;

Step 110: The Search Engine **68** locates relevant documents and formats a response to the user **56**.

Additionally, the following steps occur either prior to, or concurrently with the retrieval:

Step 112: A program (called a crawler or a spider **64**) collects documents **66** from various sources (including the World Wide Web);

Step 114: The documents **66** are processed to determine which terms are contained in them and the corresponding entries are added to the reverse index.

Note that the search engine structure **76** of **Figure 10** is similar to an existing search engine structure **54** of **Figure 8** (except for the inclusion of the QR module **78**). In particular, an existing search engine **68** and an existing reverse index **62** may be used.

D. Categorization Using Word Sense Disambiguation

Categorization was previously defined to be the identification of the topic(s) contained in a document. In the context of the present invention, categorization generally comprises the steps of:

a) Determining which nodes in a disambiguated document are the most important
b) Assigning numerical measures of importance to these nodes.

These two steps are accomplished by providing an exact measure of the amount of information that each node contributes to the document. Claude Shannon's Information Entropy Theory provides such a measure.

The entropy for a particular node x, represented by h_{nx}, may be defined as:

Eqn 7: $$h_{nx} = -\log(c_{nx}/G)$$

Where c_{nx} is the count associated with node x as defined earlier, and G is the sum of c_{nx} for all nodes.

The entropy for an edge between node x and node y, represented as $h_{ex,y}$, may be defined as:

Eqn 8: $$h_{ex,y} = -\log(c_{ex,y}/G)$$

Where $c_{ex,y}$ is the count associated with the edge as defined above, and G is the sum of c_{nx} for all nodes.

Thus, these entropies can be easily computed from existing information. The entropies for nodes and edges may be derived from the counts at the same time the probabilities for the nodes and edges are computed.

Once the entropies for each node and edge are known, then the entropy (i.e. amount of information contributed) for a particular instance of node x in a document may be approximated as:

Eqn 9: $$H(x) = h_{nx} - (h_{ex,o} + h_{ex,p} + \ldots + h_{ex,q})/c$$

Where nodes *o*, *p* ... *q* are senses in close proximity to node *x* that are semantically related to node *x*, and *c* is the number of such nodes.

The entropy that all instances of node x (and therefore the concept represented by node x) contribute to the document is simply the sum of the entropies for each instance of node x as described above. This entropy, expressed as a percentage of the sum of the entropies for all nodes present in the document represents the relative importance of each node. With a numerical way to detect the most important nodes in the document, the present invention is able to separate the most important nodes and consider these to be the topics with which the document is concerned. The least important nodes can be considered as incidental and therefore ignored.

This system and method of categorization using WSD may be used in a number of embodiments, several of which are described below. One skilled in the art, however, will appreciate that the categorization method of the present invention may be applied in embodiments beyond those described herein.

FIRST CATEGORIZATION EMBODIMENT

The categorization system and method may be used to group together documents that deal with more or less the same topics. Such groups are often hierarchical and are called directories (e.g. the Yahoo directory of documents which consists of approximately 11,000 groups/categories). **Figure 11** shows the general steps in such a method.

In **Step 120**, a document **30**, comprised of terms $t_1 \ldots t_n$, is first processed through WSD to determine the senses **80** contained in it, $s_1 \ldots s_n$.

In **Step 122**, the senses **80** are processed through categorization to determine the relative importance **82** of each node that is present in the document, $r_o \ldots r_q$.

Finally, in **Step 124** a set of rules (similar to a rule-based system) may be used to determine which group or groups **84** the document **30** should be inserted into. The rules used to perform the grouping may be obtained manually, or may be created/adjusted automatically to simulate an existing directory by performing Probabilistic Analysis on the correlation between the topics in documents in the directory and the group these documents are placed in. Such systems are very useful in any large repository of information such as enterprise portals, libraries of publications, news feeds etc.

SECOND CATEGORIZATION EMBODIMENT

The categorization system and method may also be used to improve retrieval. Since categorization can determine which are the important topics in a document, there is no need to include all topics in the reverse index **62** of a search engine **68**. Alternatively, one may include only the relevant ones and ignore the incidental topics. **Figure 12** illustrates such a system and the step to using the system.

In **Step 126**, a spider **64** fetches a document **66**.

In **Step 128**, the fetched document **66** is passed to a WSD module **4** to determine the senses for each term in the document.

In **Step 130**, the fetched document **66** is passed to a categorization module **86** to determine the important nodes of the document **66**. Only the important nodes are included in the modified reverse index **70** in **Step 132**. Additionally, the relative importance of each node is entered into the modified reverse index for the purpose of ranking the documents once they are returned. This is typically much more accurate than the simple word counts that are in use in existing systems.

E. Alternative Embodiments

The present invention contains numerous features that may be varied to yield alternative embodiments. Some possible variants are listed below.

Numerical Metrics for WSD Other Than Probability

In a preferred embodiment, the present invention employs equivalent metrics instead of probabilities. In particular, one can define the following:

$$w_{nx} = -\log(p_{nx})$$

and replace **Equations 1, 2,** and **3** by the following respectively:

Eqn 10: $w_{ex,y} = -\log(p_{x,y}) - w_{nx} - w_{ny}$

Eqn 11: $W(x) = w_{nx} - (w_{ex,o} + w_{ex,p} + \ldots * w_{ex,q})/c$

Eqn 12: $W_{interpretation} = W(s_1) + W(s_2) + W(s_3) + \ldots + W(s_n)$

As a result, instead of attempting to find the interpretation with the maximum $P_{interpretation}$, the system attempts to find the interpretation with the minimum $W_{interpretation}$. Basing **Equations 10, 11** and **12** on **Equations 1, 2** and **3** guarantees that the two interpretations will be the same. However, the new metric has two advantages. First, most of the multiplications and divisions have been replaced by additions and subtractions respectively, which are faster to implement on a computer thus making the program run faster. Secondly, probabilities are numbers between zero and one and as they are multiplied together, they quickly turn to very small numbers leading to the possibility of loss of accuracy due to rounding errors. The numbers in the new metric are bigger and they are added instead of being multiplied so the loss of accuracy is minimized.

Note that any equivalent metric can be used as long as it guarantees to select the same interpretation as the probability metric.

Edge Propagation

In cases where a document to be disambiguated consists of very few terms (such as the case of user-typed queries), there is very little context from which to derive the sense of each term. Therefore, it may be possible that there are no semantic edges that directly link nodes for a term with nodes of any other term in the document, and that term is then impossible to disambiguate from context. However, the present invention uses indirect relationships to disambiguate such documents.

Consider the query "tree cork" with the shown in **Figure 13**. While there may be no edge relating the nodes corresponding to the term "tree" **(88a, 88b** and **88c)** with any of the nodes corresponding to the word "cork" **(90a, 90b** and **90c)**, there are edges **92** relating the nodes of the term "tree" with the nodes of the term "bark" **(94,96)**, and the nodes of the term "bark" **(94,96)** with the nodes of the term "cork". The present invention can create multiple "virtual" edges **98** to represent these relationships, and the probabilities associated with them. The virtual edges may be used to disambiguate terms in the original document that may only be connected by such virtual edges. Although depicted is a network in which the virtual edge has only one intervening set of nodes **(94,96)**, it can easily be seen that this method may be expanded to apply to an arbitrary number of intervening nodes.

Note that the virtual edge probabilities are equal to the product of the probabilities of all semantic edges that make up the path. In particular, the probability for virtual edge **98** is the product of the probabilities associated with semantic edge **101** and semantic edge **103.** It can be seen that the effect of including virtual edges in the computation that results in interpretation selection is equivalent to the effect of including the intervening terms in the original document, performing WSD as usual and then removing the senses for the intervening terms from the result. This edge propagation method is shown in **Figure 14**.

Other Heuristics for Interpretation Selection

In other embodiments, alternative algorithms are used to perform interpretation selection. Described in detail above is the preferred modified Gradient Descent algorithm but obviously any exact or approximate heuristics (such as the ones listed in section A) can be used. The heuristic employed can affect dramatically both the accuracy of the system as well as the speed with which it produces a possible solution.

Removing Edges That Are Not Truly Relevant

Because of the use of probabilities (or equivalent metrics thereof) in the WSD process **400**, it is possible to have edges in the network that are not truly relevant: the probabilities for these edges are not favorable to the nodes related by the edge so the presence of the edge does not affect the outcome of the WSD process. It is therefore possible in some embodiments to remove such edges (and adjust the probabilities of remaining edges accordingly if necessary) to reduce the size of the network. Smaller networks operate faster and consume less memory in the computer.

Alternative Methods of Adding Edges

Described above is a method of adding edges to a network to augment any that correspond to known semantic relationships. This method was based on correlation analysis. Since these edges do not necessarily have to correspond to true semantic relationships, alternative embodiments of the present invention employ any method that can distinguish between edges that *could* correspond to semantic relationships (including random guessing or creating all possible edges). The accuracy with which it detects edges that correspond to true semantic relationships does not affect the accuracy of the WSD results; the results depend only on the accuracy of the calibration process. However, higher accuracy in detecting true semantic relationships reduces the number of unnecessary edges (i.e. edges that do not correspond to true semantic relationships) and increases the number of useful edges (i.e. edges that do correspond to true semantic relationships).

Alternative Methods of Grouping Documents

The grouping of documents after categorization does not necessarily have to be implemented using a rule-based system. In alternative embodiments, the nodes in the network are arranged into a directory hierarchy themselves. Under such a scheme the node representing Business could be designated as a category in the hierarchy, the nodes representing Health Services, Financial Services, Entertainment etc. could be designated as sub-categories and placed within the Business category, the node representing Banking could be placed within the Financial Services sub-category and so on. Given such a hierarchy, the nodes present in the topics of a given document after categorization (such as the node representing Banking) could be used to place it into the groups represented by the nodes (in this case the sub-category related to the node for Banking). Other possible schemes include the use of a probabilistic model, a neural network, or any other existing pattern-matching technique, to replicate a particular categorization scheme created either by manual effort or any other means.

Automated Document Translation

One of the advantages of semantic WSD techniques is that they are language invariant (i.e. they can operate in any language). This is because the main information they use to perform WSD is semantic relationships which are themselves language invariant: the noun "dog" is semantically related to the verb "bark" no matter which language is used to express the fact.

In another embodiment, the current invention provides a system and method of automated document translation. For the purposes of automatically translating documents from one language to another, it is absolutely necessary to know the sense of each term in the source document. For example, while the verb "bark" and the noun "bark" are spelled the same way in English, this is not so for German. So, if it were necessary to translate a document containing the word "bark" from English to German, one would have to know whether it refers to the verb or the noun.

The combination of the above makes semantic WSD techniques particularly useful for automatic document translation.

Natural Language Processing

In another embodiment, the present invention provides a system and method for enhancing natural language processing. Existing language parsing techniques often only extract the context that is contained in the structure of a sentence. Quite often, this is not enough to resolve all possible ambiguities (even within the structure of the sentence). Standard Natural Language Processing techniques can be enhanced with semantic WSD techniques to resolve as many ambiguities as possible thus enhancing the effectiveness of Natural Language Processing.

In such a system, a Natural Language Processing module could extract the sentence structure. Determining the type of each term (verb, noun, adjective, etc.) will reduce the number of possible senses that a WSD module will have to consider. At the same time, a Word Sense Disambiguation module could determine the sense of each term, thus providing information that could be used to resolve ambiguities that the Natural Language Processing system could not resolve in the first place. In effect, the two techniques are complementary.

Grammar and Syntax Checking

In another embodiment, the present invention provides a system and method for improving existing grammar and syntax checking systems. Quite often it is possible to mistype words in such a way that they match another valid word and therefore do not get pointed out by spell-checking systems. A common example of this is the words "form" and "from". A mistyped word of this type will quite often not match the context of the remainder of the sentence (often not matching the structure of the remainder of the sentence either). The WSD method described herein attempts to select an interpretation of a document by making the context of the document as coherent as possible. Therefore, the proposed system could be designed so as to look for senses corresponding to alternative spellings as well as the senses corresponding to the given spelling, and determine if any permutation of the spelling would yield a better interpretation. In particular, for the example above, the WSD process could include all the senses of the term "form" even though it is the term "from" that appears in the document. If the interpretations containing one or more senses corresponding to "form" have higher probabilities than the ones corresponding to "from", then the system could automatically suggest the correction to a human user. Such a system would be able to catch errors that can only be detected by the context of the sentence and would therefore evade traditional spell checking and grammar-checking systems.

Dynamic Personalization

In another embodiment, the present invention provides a method and system for dynamic personalization, the process of customizing the documents presented to a user according to what the user preferences (whether express or implied) at any given time.

A traditional application for dynamic personalization is Electronic Advertising on the World Wide Web. On a number of commercial sites (such as Yahoo and Lycos), "banner" advertisements (graphics that advertise a product or service) are placed at the top of the displayed HTML page with links that will transfer the user to the company Web site if and when she clicks on the banner. As in traditional advertising channels, the goal is to present the particular banner to the right demographic to get maximum benefit from the advertising. Unlike the traditional channels, one has the option to change which advertisement is displayed to which individual user according to what the user seems to be interested at the time. So a user who is currently reading a page on high-performance cars would be presented with advertisements from high-performance car companies, while a user reading a page on interior decoration would be shown advertisements for home decorating products.

Thus, the document (in this case the advertising banner) is dynamically changed to reflect the interests of the user at any given time (i.e. personalized). Note that such a system is not necessarily restricted to advertising: the document that is being personalized could be any document at all while the information used to determine the current interests of the user could include a wide range of information (such as records on age, location, sex, occupation, previously selected interests, the path the user followed to the current page, etc.). Thus, dynamic personalization can in general be used to offer to the user information that may be relevant to his current needs.

In order to be able to implement a system such as the one described above, it is first necessary to know what the current needs of the user are. In the simple case where only the document currently being viewed by the user is used to determine current needs, the document currently being viewed can be processed through WSD categorization as described herein and the topics it deals with used as an indication of the interests of the user. If one would like to take into account previously declared information, one can also express this as topics that the user may be interested in.

Given the interests of the user, the remaining task is to identify or create the dynamically generated document to be offered to the user. Generally, this is a process of identification of the most appropriate document out of a class of documents. This process is in fact retrieval and the WSD based retrieval methods described herein can be employed to return the most relevant dynamic document. In addition, sometimes the set of documents available for dynamic display are limited (such as in the case of advertising where there is a relatively small number of banners – at most a few thousand).

Establishing the relevance of every set of user interests to such a set of documents is hard since there may be no exact matches between the documents to be displayed and the user interests. In such cases, the semantic relationships inherent in the system described herein can be used to extrapolate the topics to more general concepts to allow for matching. For example, if the user interests are on car racing but there is no advertising banner related to racing, the semantic relationship between cars and racing could be used to display a banner for high-performance cars, or the semantic relationship between racing and sports could be used to display a banner for a sports station on cable.

Customer Relationship Management Applications

In another embodiment, the present invention provides a system and method for customer relationship management (CRM) applications, which focus on providing enhanced service and support to customers. Therefore, it is often very similar to dynamic personalization in that the system must determine the needs of the user and service them in the most efficient manner possible.

For example, consider the issue of cross-selling items (a process where the user purchases an item such as wine glasses and is offered the option to purchase a related item such as a cork-screw). In this case, determining possible user needs consists (in part) of determining the topics of the document the user is currently viewing (in the example above the product description for the wine glasses). This can be performed by WSD based categorization. Similarly, determining possible ways to satisfy a user need (such as items to cross-sell) consists (in part) of determining related documents/actions (in the example above, the product description for the cork-screw). Other examples of CRM applications are providing the user with electronic copies of Operating Manuals for products purchased, Frequently Asked Questions for an item purchased, etc. In all such cases the possible user needs can be determined (in part) by using WSD based categorization and possible solutions determined by using the semantic relationships inherent in the method, or by performing retrieval on the particular topics to extract relevant documents from a corpus containing known solutions.

Automatic Hyper-Linking of Documents

In yet another embodiment, the present invention provides a system and method of automatically hyper-linking documents. Quite often, it is important to hyper-link particular topics presented in a document to other documents that may contain substantial background information on the topic. A typical example is encyclopedias (where articles are often cross-linked) or news articles (where mentions of earlier incidents are cross-linked to articles describing those incidents or articles describing related incidents).

This process is often too laborious to undertake manually. By performing WSD based categorization on a large database of documents, one can determine which topics are analyzed in depth in each document. One can then create links in other documents that point to these as appropriate.

Automatic Summarization

In another embodiment, the present invention provides a system and method of automatic summarization of documents. The Information Entropy based method of performing categorization can also determine which sentences within a document provide the most information. This can be used to extract the most information-rich sentences out of a document to automatically create a summary. Using the same methods, if a user is interested in a particular item of information contained in a document (e.g. information discussed in a particular paragraph), during retrieval the system can point out the individual items of information rather than expect the user to read through the whole document.

Automatic Speech Recognition

In another embodiment, the present invention provides a system and method of improving speech recognition systems. Speech recognition is a challenging application that has the same problems of ambiguity associated with it as WSD. Quite often, words are pronounced the same (or pronounced with accents that make them sound the same as some other word) but typical methods of speech recognition have failed to resolve such ambiguities effectively. WSD can be used in conjunction with existing speech recognition techniques to resolve such ambiguities.

Such a system would operate as follows: A traditional speech recognition system would translate speech to the possible nodes that each word could refer to. Then a Word Sense Disambiguation module of the type described above can be used to select one node for each word thus resulting in the final interpretation of the spoken input. Once a final interpretation is selected, the list of nodes in the interpretation can be converted to a text document for further processing.

Optical Character Recognition (OCR)

In another embodiment, the present invention provides a system and method for improving OCR systems. Once again, OCR suffers from the issue of ambiguities – sometimes it is hard to distinguish between alternatives for one single character (for example, the hand-written versions of "c" and "e" tend to look very similar). Existing methods (mainly based on Markov models) often fail to provide the required accuracy, which is why OCR has never gained substantial popularity. A system similar to the one for speech recognition could be established which could use Word Sense Disambiguation to determine the correct version of the word (and therefore the correct spelling) at each point.

Other embodiments of the invention will be apparent to those skilled in the art from a consideration of the specification or practice of the invention disclosed herein. It is intended that the specification and examples be considered as exemplary only, with the true scope and spirit of the invention being indicated by the following claims.

What is claimed is:

Claims

1. A method of performing word sense disambiguation on a document, comprising the steps of:

providing a network comprised of nodes corresponding to senses, edges corresponding to semantic relationships between the senses, and the probabilities of the nodes and the edges;

receiving a document;

converting the document into a list of terms;

applying a stemming algorithm to the list of terms to obtain a stem list for each term in the list of terms;

looking up in the network each stem in the stem list to determine all senses possibly referring to each stem, thereby obtaining a set of senses for each stem that could represent the corresponding term;

applying a heuristic to select likely interpretations for each set of senses, an interpretation consisting of a list of senses wherein each sense in the list has been selected to correspond to a term out of all the possible senses for that term;

calculating the probability of each interpretation being the correct interpretation; and

returning the most likely interpretation of the document.

2. The method of claim 1, wherein the heuristic is an exact heuristic.

3. The method of claim 1, wherein the heuristic is an approximate heuristic.

4. The method of claim 1, wherein applying a heuristic further comprises the steps of:

   for each sense set s_j such that *i* is not equal to *j*, selecting as an initial sense for sense set s_i the sense that results in a maximum calculated probability of a correct interpretation, using in the probability calculation only cross-connected pairs consisting of a node from sense set s_i and a node from sense set s_j, thereby obtaining a current interpretation;

   if initial senses for each sense set have been selected, proceeding to the next step, otherwise performing the previous step on the next sense set;

   for each sense in sense set s_i, adjusting the initial sense selected for sense set s_i for the current interpretation to be the sense that results in the maximum calculated probability of a correct interpretation, using in the calculation all senses in the current interpretation;

   if all sense sets have been adjusted, proceeding to the next step, otherwise performing the previous step on the next sense set; and

   repeating the previous two steps until no more adjustments are required.

5. The method of claim 1, wherein applying a heuristic further comprises the steps of:

   for each sense set s_j such that *i* is not equal to *j*, selecting as an initial sense for sense set s_i the sense that results in a maximum calculated probability of a correct interpretation, using in the probability calculation only cross-connected pairs consisting of a node from sense set s_i and a node from sense set s_j, thereby obtaining a current interpretation;

if initial senses for each sense set have been selected, proceeding to the next step, otherwise performing the previous step on the next sense set;

for each sense in sense set s_i, adjusting the initial sense selected for sense set s_i for the current interpretation to be the sense that results in the maximum calculated probability of a correct interpretation, using in the calculation all senses in the current interpretation;

if all sense sets have been adjusted, proceeding to the next step, otherwise performing the previous step on the next sense set; and

repeating the previous two steps until a predefined number of iterations has been performed.

6. A method of performing word sense disambiguation on a document, comprising the steps of:

providing a network comprised of nodes corresponding to senses, edges corresponding to semantic relationships between the senses, and the probabilities of the nodes and the edges;

receiving a document;

converting the document into a list of terms;

applying a stemming algorithm to the list of terms to obtain a stem list for each term in the list of terms;

looking up in the network each stem in the stem list to determine all senses possibly referring to each stem, thereby obtaining a set of senses for each stem that could represent the corresponding term;

applying a heuristic to select likely interpretations for each set of senses, an interpretation consisting of list of senses wherein each sense in the list has been selected to correspond to a term out of all the possible senses for that term;

calculating the likelihood that each interpretation is the correct interpretation by defining node weights and edge weights in terms of logarithms of node and edge probabilities, defining sense weights in terms of corresponding node weights and edge weights, and summing the sense weights corresponding to each interpretation; and

returning the most likely interpretation of the document by selecting the interpretation whose sense weight sum is a minimum.

7. The method of claims 1 or 6, further comprising the step of:
removing irrelevant edges from the network.

8. A method of automatically creating a network comprised of nodes, edges, and their respective probabilities, comprising the steps of:
converting a set of input files containing known senses for terms and known semantic relationships between the senses into a set of commands a computer uses to create nodes corresponding to the known senses and edges corresponding to the known semantic relationships in a network;

creating edges corresponding to synonyms extracted from the input files, homonyms identified by comparing the spelling of the terms that correspond to the nodes;

initializing for each node a node count equal to a value large enough not to be easily swayed by statistical variations of a corpus sample;

initializing for each edge an equal edge count value except for edges corresponding to known semantic relationships which are initialized to substantially higher edge count values;

assigning probabilities to the nodes and edges by determining the relative frequency of each term compared to its homonyms, and the relative frequency of a particular pair of nodes corresponding to an edge compared to all other edges between nodes corresponding to the same terms as the particular pair of nodes;

creating automatically additional edges; and

word sense disambiguating a large corpus of randomly selected documents to adjust the probabilities of the nodes and edges, thereby obtaining a finalized network.

9. The method of claim 8, wherein the input files are lexicographer files.

10. The method of claim 8, wherein the input files are any files containing known semantic relationships.

11. The method of claim 8, wherein creating automatically additional edges further comprises creating automatically additional edges corresponding to

and all pairs of nodes identified through correlation analysis as potentially semantically related.

12. The method of claim 11, wherein pairs of nodes identified through correlation analysis comprises pairs of nodes whose correlation coefficients are greater than a predetermined threshold.

13. The method of claim 8, wherein creating automatically additional edges further comprises creating all possible edges.

14. The method of claim 8, wherein creating automatically additional edges further comprises creating edges randomly.

15. A method of retrieval using word sense disambiguation, comprising the steps of:

receiving a query comprised of a set of keywords;

word sense disambiguating the keywords to identify the senses corresponding to the keywords;

entering the senses into a search engine; and

retrieving via the search engine entries from a sense-based reverse index.

16. The method of claim 15, further comprising the step of:
returning to a user a response corresponding to the retrieved entries.

17. The method of claim 15, further comprising the step of:

providing a sense-based reverse index comprised of entries corresponding to senses resulting from word sense disambiguating documents collected by a crawler or a spider.

18. A method of reformulating a query using word sense disambiguation, comprising the steps of:

receiving a query comprised of a set of keywords;

word sense disambiguating through a semantic network the keywords to identify the senses corresponding to the keywords; and

reformulating the query to include the identified senses while removing ambiguity.

19. A method of retrieval using word sense disambiguation, comprising the steps of:

receiving a query comprised of a set of keywords;

word sense disambiguating the keywords to identify the senses corresponding to the keywords;

reformulating the query to include the identified senses while removing ambiguity;

entering the reformulated query into a search engine; and

retrieving via the search engine entries from a reverse index.

20. The method of claim 19, further comprising the step of:

arranging the presentation of documents corresponding to the retrieved entries by their relative importance to the identified senses.

21. The method of claim 19, further comprising the step of:

returning to a user a response corresponding to the retrieved entries.

22. The method of claim 19, further comprising the step of:

providing a reverse index to which has been added entries corresponding to terms contained in documents collected by a crawler or spider.

23. A method of categorization using word sense disambiguation, comprising the steps of:

providing a network comprised of nodes corresponding to senses, and edges corresponding to semantic relationships between the senses;

receiving a document comprised of a plurality of terms;

word sense disambiguating the document to obtain nodes and edges corresponding to senses and semantic relationships of the document;

determining the relative importance of each node by calculating the entropy of each node, thereby obtaining a set of node entropies; and

applying grouping rules to the set of node entropies to determine into which topical group or groups the document should be inserted.

24. The method of claim 23, wherein the grouping rules are obtained manually.

25. The method of claim 23, wherein the grouping rules are created automatically by performing probablistic analysis on the correlation between the set of node entropies and the topical group or groups.

26. A method of retrieval using word sense disambiguation, comprising the steps of:

providing a sense-based reverse index comprised of entries corresponding to only the most important senses as identified per the method of claim 23 and resulting from word sense disambiguating documents collected by a crawler or a spider;

receiving a query comprised of a set of keywords;

word sense disambiguating the keywords to identify the senses corresponding to the keywords;

entering the senses into a search engine; and

retrieving via the search engine entries from the sense-based reverse index.

27. A method of propagating virtual edges for use in word sense disambiguating a document, comprising the steps of:

providing a network comprised of nodes corresponding to senses, edges corresponding to semantic relationships between the senses, and the probabilities of the nodes and the edges;

receiving a document;

converting the document into a list of terms;

locating intervening terms in the network between terms corresponding to terms in the document for which no directly connected semantically related pair exists in the network;

defining virtual edges between the nodes corresponding to the indirectly related document terms, wherein the virtual edge probabilities are equal to the product of the probabilities of all semantic edges connecting the nodes;

word sense disambiguating the document as in claim 1 including the intervening terms;

removing the nodes corresponding to the intervening terms.

28. A method of using word sense disambiguation in automated document translation, comprising the steps of:

providing a first network and a second network each comprised of nodes corresponding to senses, and edges corresponding to semantic relationships between the senses, wherein the first network is expressed in a first language, and the second network is expressed in a second language;

receiving a document in the first language;

word sense disambiguating the document using the first network to obtain the most likely interpretation of the document independent of the first language;

reverse word sense disambuating the language independent interpretation with the second network, thereby obtaining a document in the second language corresponding to the most likely interpretation of the document in the first language.

29. A method of using word sense disambiguation in natural language processing, comprising the steps of:

providing a network comprised of nodes corresponding to senses, and edges corresponding to semantic relationships between the senses;

receiving a document;

extracting the document sentence structure through natural language processing, thereby obtaining a set of terms with a reduced number of possible senses; and

word sense disambiguating the set of terms to obtain the most likely interpretation.

30. A method of using word sense disambiguation in grammar and syntax checking, comprising the steps of:

providing a network comprised of nodes corresponding to senses, and edges corresponding to semantic relationships between the senses;

receiving a document;

identifying mispelled or improperly used terms by word sense disambiguating the document and identifying senses with low resulting probabilities.

31. The method of claim 30, further comprising the step of:

suggesting a correction to the mispelled or improperly used term by determining whether any permutation of the term results in a better interpretation.

32. A method of using word sense disambiguation in dynamic personalization, comprising the steps of:

providing a network comprised of nodes corresponding to senses, and edges corresponding to semantic relationships between the senses;

receiving a document currently being viewed by an online computer user;

categorizing the document through word sense disambiguation to identify topics and thereby the current needs of the computer user;

retrieving using word sense disambiguation the most relevant static or dynamic documents reflecting the identified topics.

33. The method of claim 32, further comprising the step of:

modifying the identified topics related to documents previously viewed by the online computer user and word sense disambiguated.

34. The method of claim 32, further comprising the step of:

extrapolating the identified topics to more general concepts by exploiting semantic relationships between the identified topics.

35. A method of using word sense disambiguation in customer relationship management, comprising the steps of:

providing a network comprised of nodes corresponding to senses, and edges corresponding to semantic relationships between the senses;

receiving a document currently being viewed by an online customer;

categorizing the document through word sense disambiguation to identify topics and thereby the current needs of the customer; and satisfying the customer by retrieving through word sense disambiguation the most relevant documents reflecting the identified topics.

36. The method of claim 35, wherein satisfying the customer further comprises:

identifying an action possibly desired by the customer by exploiting semantic relationships between the identified topics.

37. A method of using word sense disambiguation in automatically hyper-linking documents, comprising the steps of:

providing a network comprised of nodes corresponding to senses, and edges corresponding to semantic relationships between the senses;

receiving a large database of documents;

performing word sense disambiguation categorization on the database, thereby identifying topics substantially referenced in the database;

receiving a document to be hyper-linked; and

linking terms in the received document to the identified topics in the database.

38. A method of using word sense disambiguation in automatic summarization of documents, comprising the steps of:

providing a network comprised of nodes corresponding to senses, and edges corresponding to semantic relationships between the senses;

receiving a document;

categorizing through word sense disambiguation the document; and identifying which sentences of the document provide the most information by measuring the sentences' entropies.

39. A method of using word sense disambiguation in automatic speech recognition, comprising the steps of:

providing a network comprised of nodes corresponding to senses, and edges corresponding to semantic relationships between the senses;

receiving from the output of a speech recognition system a set of terms representing each word spoken into the system; and

identifying misspelled or improperly detected terms by word sense disambiguating the set of terms and identifying senses with low resulting probabilities.

40. A method of using word sense disambiguation in optical character recognition, comprising the steps of:

providing a network comprised of nodes corresponding to senses, and edges corresponding to semantic relationships between the senses;

receiving from the output of an optical character recognition system a set of terms representing each word scanned into the system; and

identifying misspelled or improperly detected terms by word sense disambiguating the set of terms and identifying senses with low resulting probabilities.

41. The method of claim 39 or 40, further comprising the step of:

suggesting a correction to the mispelled or improperly detected terms by determining whether any permutation of the terms results in a better interpretation.

42. A system for performing word sense disambiguation on a document, comprising

memory means to store a network and computer-executable process steps;

a network comprised of nodes corresponding to senses, edges corresponding to semantic relationships between the senses, and the probabilities of the nodes and the edges; and

a processor that executes computer-executable process steps so as
to receive a document;
to convert the document into a list of terms;
to apply a stemming algorithm to the list of terms to obtain a stem list for each term in the list of terms;
to look up in the network each stem in the stem list to determine all senses possibly referring to each stem, thereby obtaining a set of senses for each stem that could represent the corresponding term;
to apply a heuristic to select likely interpretations for each set of senses, an interpretation consisting of a list of senses wherein each sense in the list has been selected to correspond to a term out of all the possible senses for that term;
to calculate the probability of each interpretation being the correct interpretation; and
to return the most likely interpretation of the document.

43. Computer-executable process steps stored on a computer-readable medium, the computer-executable process steps comprising:

code to receive a document;

code to convert the document into a list of terms;

code to apply a stemming algorithm to the list of terms to obtain a stem list for each term in the list of terms;

code to look up in a network each stem in the stem list to determine all senses possibly referring to each stem, thereby obtaining a set of senses for each stem that could represent the corresponding term, the network further comprised of nodes corresponding to senses, edges corresponding to semantic relationships between the senses, and the probabilities of the nodes and the edges;

code to apply a heuristic to select likely interpretations for each set of senses, an interpretation consisting of a list of senses wherein each sense in the list has been selected code to correspond to a term out of all the possible senses for that term;
code to calculate the probability of each interpretation being the correct interpretation; and

code to return the most likely interpretation of the document.

44. A system for automatically creating a network comprised of nodes, edges, and their respective probabilities, comprising:

memory means to store a network and computer-executable process steps;

a processor that executes computer-executable process steps so as to convert a set of input files containing known senses for terms and known semantic relationships between the senses into a set of commands a computer uses to create nodes corresponding to the known senses and edges corresponding to the known semantic relationships in a network;

to create edges corresponding to synonyms extracted from the input files, homonyms identified by comparing the spelling of the terms that correspond to the nodes;

to initialize for each node a node count equal to a value large enough not to be easily swayed by statistical variations of a corpus sample;

to initialize for each edge an equal edge count value except for edges corresponding to known semantic relationships which are initialized to substantially higher edge count values;

to assign probabilities to the nodes and edges by determining the relative frequency of each term compared to its homonyms, and the relative frequency of a particular pair of nodes corresponding to an edge compared to all other edges between nodes corresponding to the same terms as the particular pair of nodes;

to create automatically additional edges corresponding to and all pairs of nodes identified through correlation analysis as potentially semantically related; and

to word sense disambiguate a large corpus of randomly selected documents to adjust the probabilities of the nodes and edges, thereby obtaining a finalized network.

45. Computer-executable process steps stored on a computer-readable medium, the computer-executable process steps comprising:

code to convert a set of input files containing known senses for terms and known semantic relationships between the senses into a set of commands a computer uses to create nodes corresponding to the known senses and edges corresponding to the known semantic relationships in a network;

code to create edges corresponding to synonyms extracted from the input files, homonyms identified by comparing the spelling of the terms that correspond to the nodes;

code to initialize for each node a node count equal to a value large enough not to be easily swayed by statistical variations of a corpus sample;

code to initialize for each edge an equal edge count value except for edges corresponding to known semantic relationships which are initialized to substantially higher edge count values;

code to assign probabilities to the nodes and edges by determining the relative frequency of each term compared to its homonyms, and the relative frequency of a particular pair of nodes corresponding to an edge compared to all other edges between nodes corresponding to the same terms as the particular pair of nodes;

code to create automatically additional edges corresponding to and all pairs of nodes identified through correlation analysis as potentially semantically related; and

code to word sense disambiguate a large corpus of randomly selected documents to adjust the probabilities of the nodes and edges, thereby obtaining a finalized network.

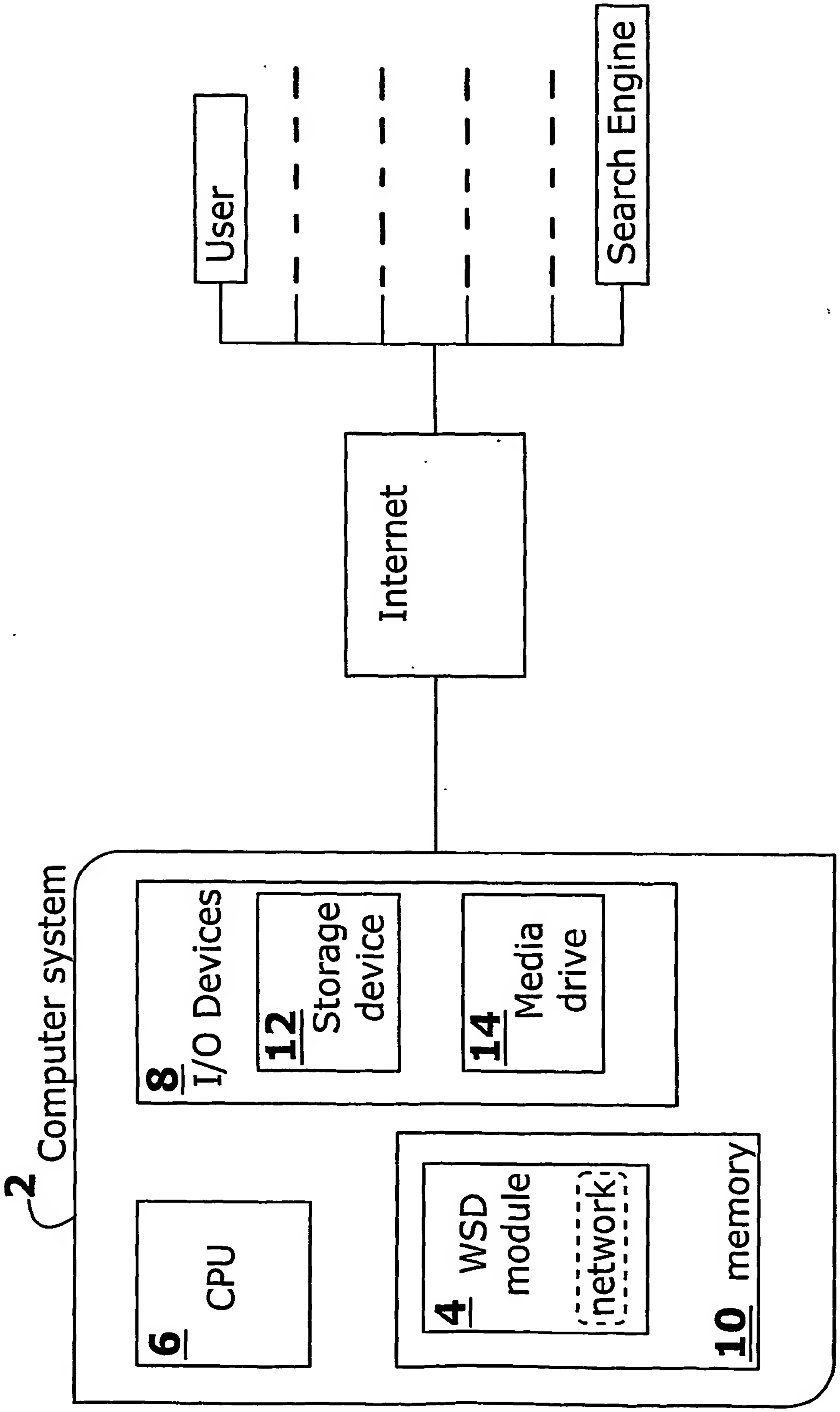

User
Search Engine
Internet
2
Computer system
6
CPU
8
I/O Devices
12
Storage device
14
Media drive
4
WSD module
network
10
memory

FIG. 1

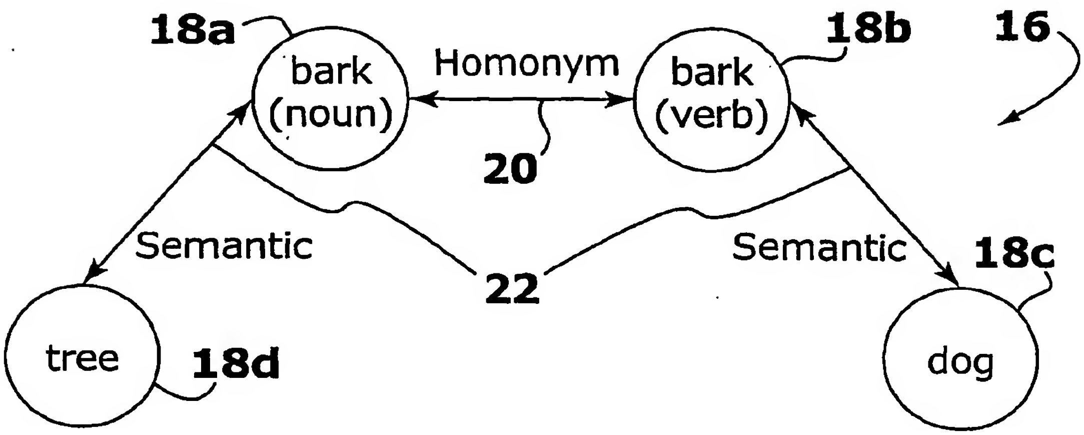

18a
bark (noun)
Homonym
bark (verb)
18b
16
20
Semantic
22
Semantic
18c
tree
18d
dog

FIG. 2

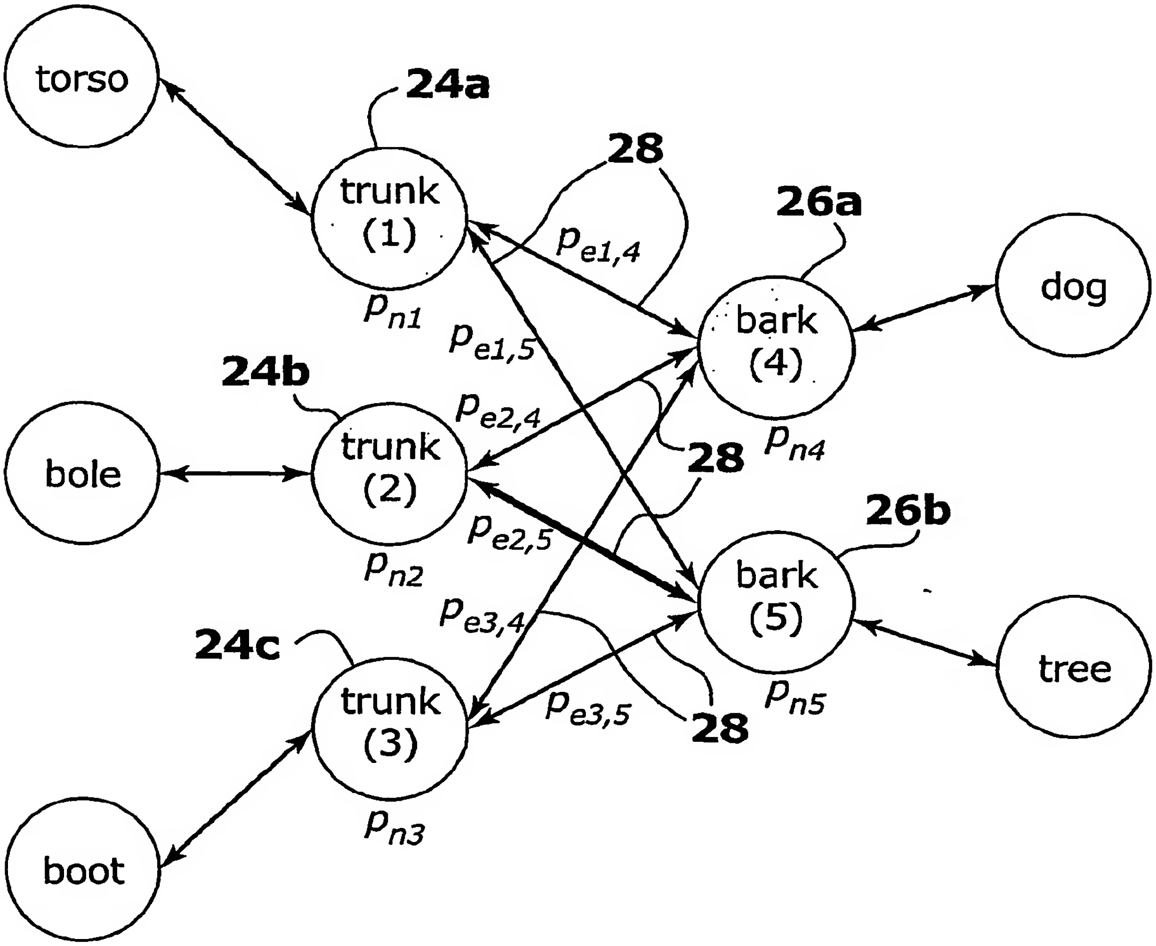

torso
24a
28
trunk (1)
$p_{e1,4}$
26a
dog
p_{n1}
$p_{e1,5}$
bark (4)
24b
$p_{e2,4}$
bole
trunk (2)
28
p_{n4}
$p_{e2,5}$
26b
p_{n2}
bark (5)
$p_{e3,4}$
24c
tree
trunk (3)
$p_{e3,5}$
28
p_{n5}
p_{n3}
boot

FIG. 3

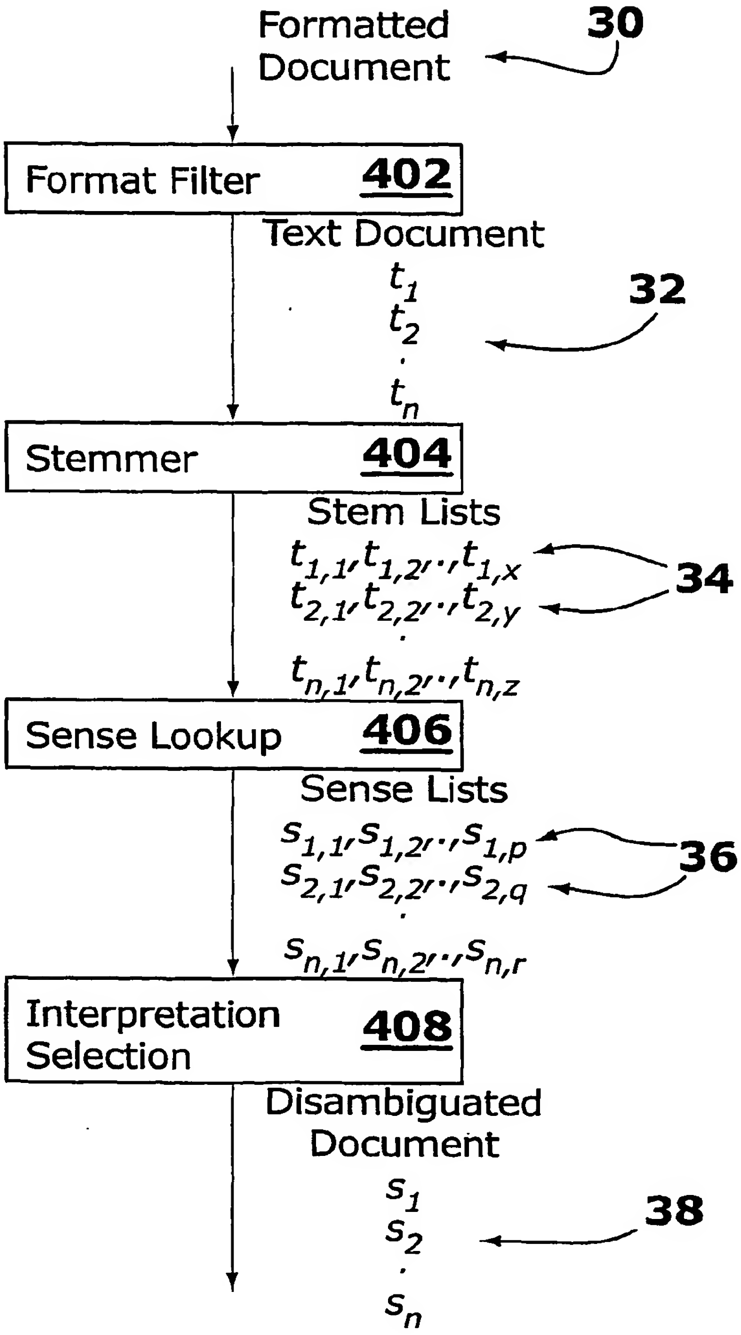

Formatted Document
30
Format Filter 402
Text Document
t_1
t_2
.
t_n
32
Stemmer 404
Stem Lists
$t_{1,1}, t_{1,2}, .., t_{1,x}$
$t_{2,1}, t_{2,2}, .., t_{2,y}$
.
$t_{n,1}, t_{n,2}, .., t_{n,z}$
34
Sense Lookup 406
Sense Lists
$s_{1,1}, s_{1,2}, .., s_{1,p}$
$s_{2,1}, s_{2,2}, .., s_{2,q}$
.
$s_{n,1}, s_{n,2}, .., s_{n,r}$
36
Interpretation Selection 408
Disambiguated Document
s_1
s_2
.
s_n
38

FIG. 4

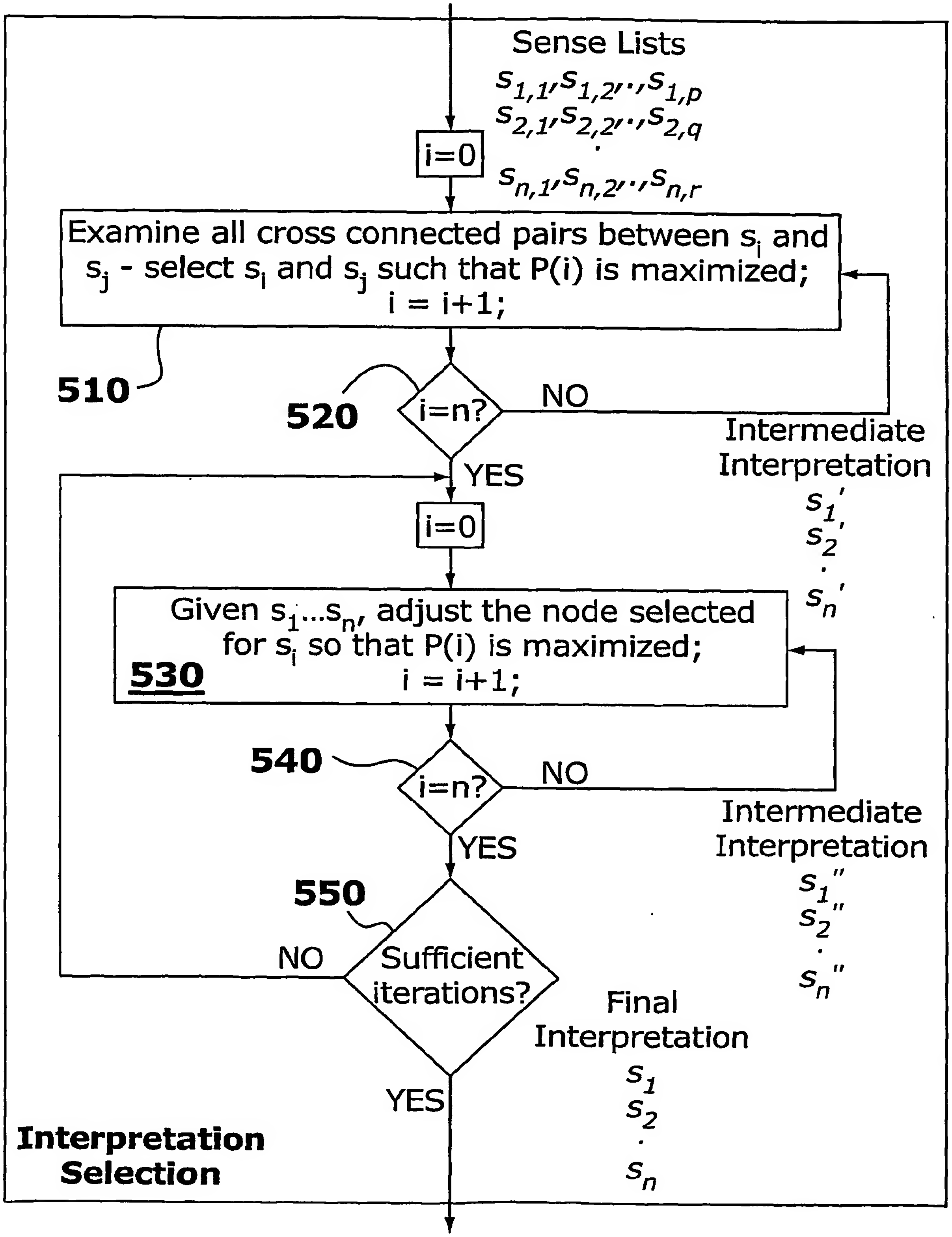

Sense Lists
$s_{1,1}, s_{1,2}, .., s_{1,p}$
$s_{2,1}, s_{2,2}, .., s_{2,q}$
.
$s_{n,1}, s_{n,2}, .., s_{n,r}$
i=0
Examine all cross connected pairs between s_i and s_j - select s_i and s_j such that P(i) is maximized; i = i+1;
510
520
i=n?
NO
Intermediate Interpretation
s_1'
s_2'
.
s_n'
YES
i=0
Given $s_1...s_n$, adjust the node selected for s_i so that P(i) is maximized; i = i+1;
530
540
i=n?
NO
Intermediate Interpretation
s_1''
s_2''
.
s_n''
YES
550
Sufficient iterations?
NO
YES
Final Interpretation
s_1
s_2
.
s_n
Interpretation Selection

FIG. 5

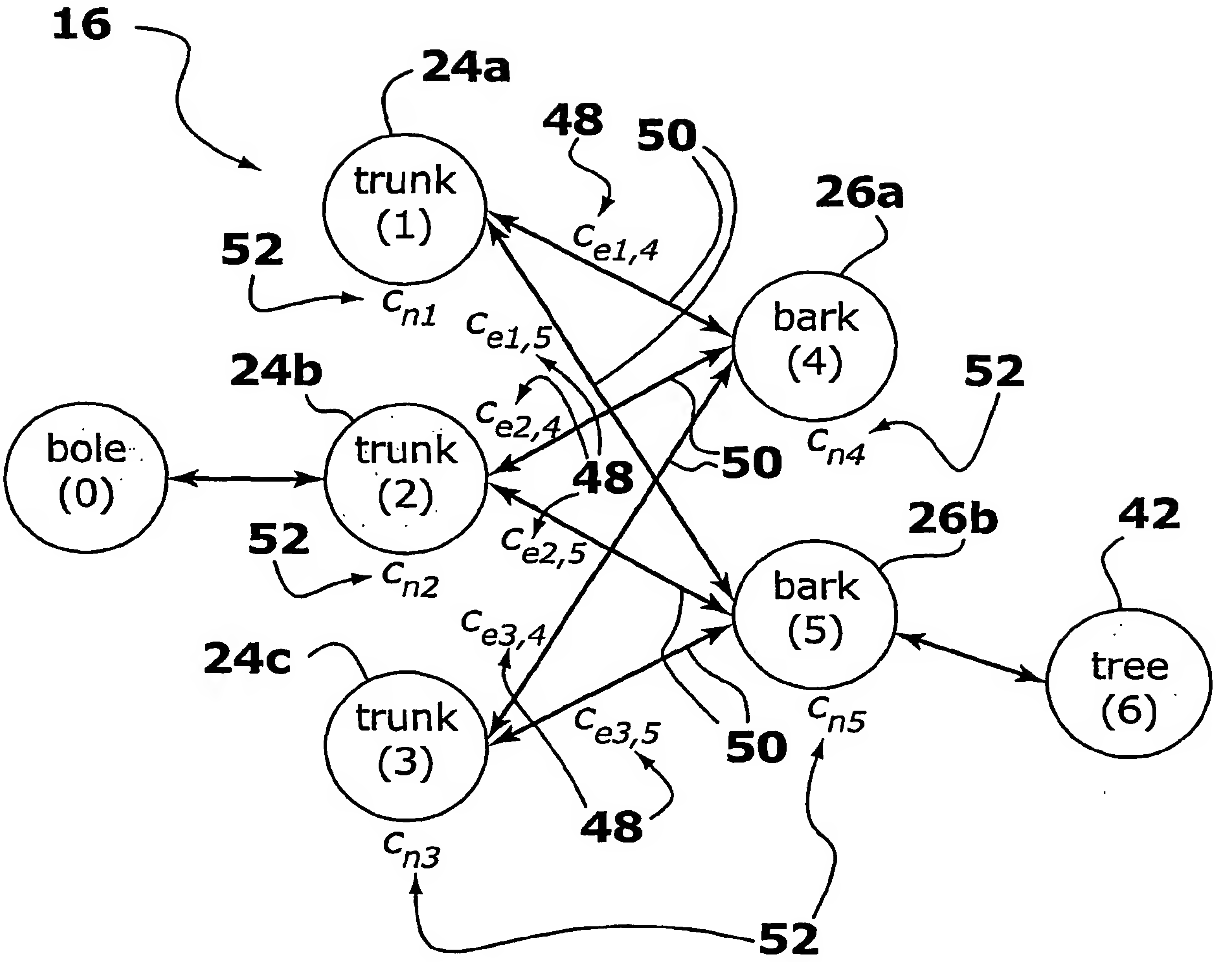

16
24a
trunk (1)
48
50
$c_{e1,4}$
26a
52
c_{n1}
$c_{e1,5}$
bark (4)
24b
52
$c_{e2,4}$
bole (0)
trunk (2)
48
50
c_{n4}
52
c_{n2}
$c_{e2,5}$
26b
42
bark (5)
tree (6)
$c_{e3,4}$
24c
trunk (3)
$c_{e3,5}$
50
c_{n5}
48
c_{n3}
52

FIG. 6

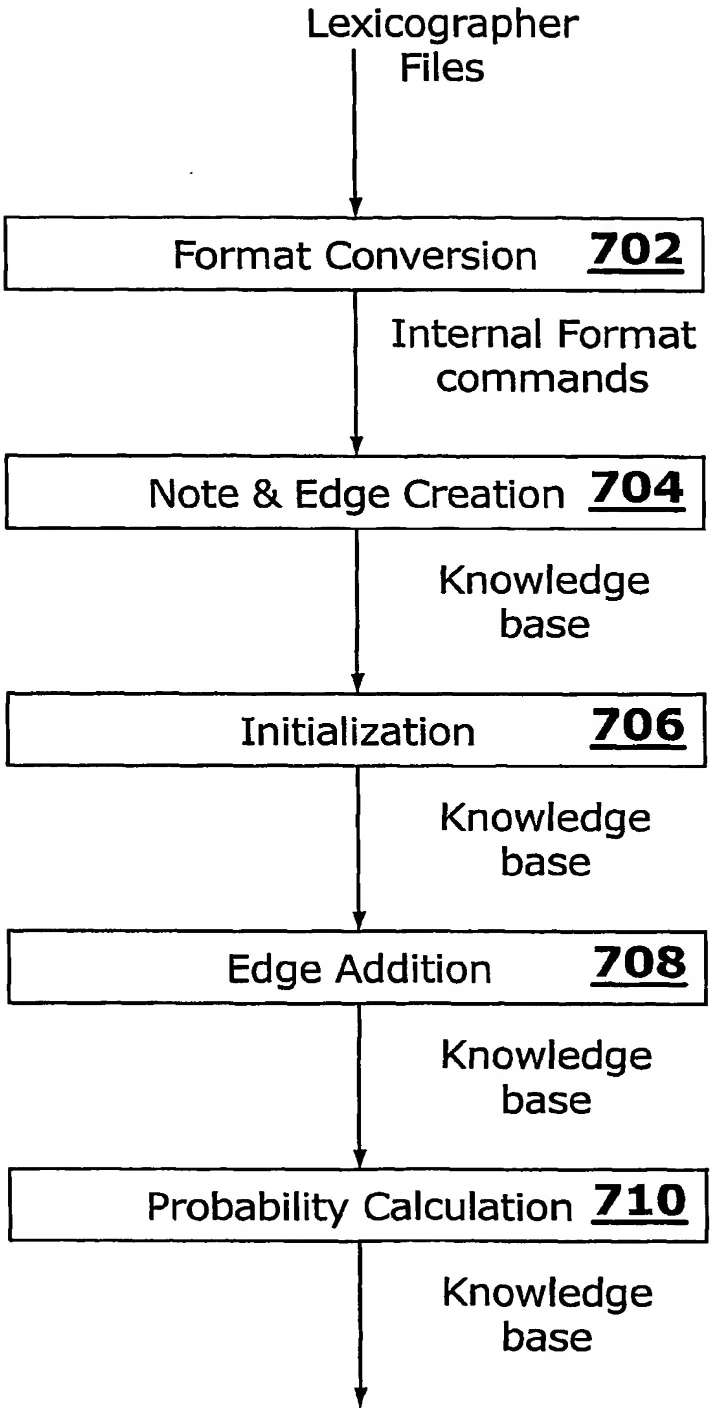

Lexicographer Files
Format Conversion 702
Internal Format commands
Note & Edge Creation 704
Knowledge base
Initialization 706
Knowledge base
Edge Addition 708
Knowledge base
Probability Calculation 710
Knowledge base

FIG. 7

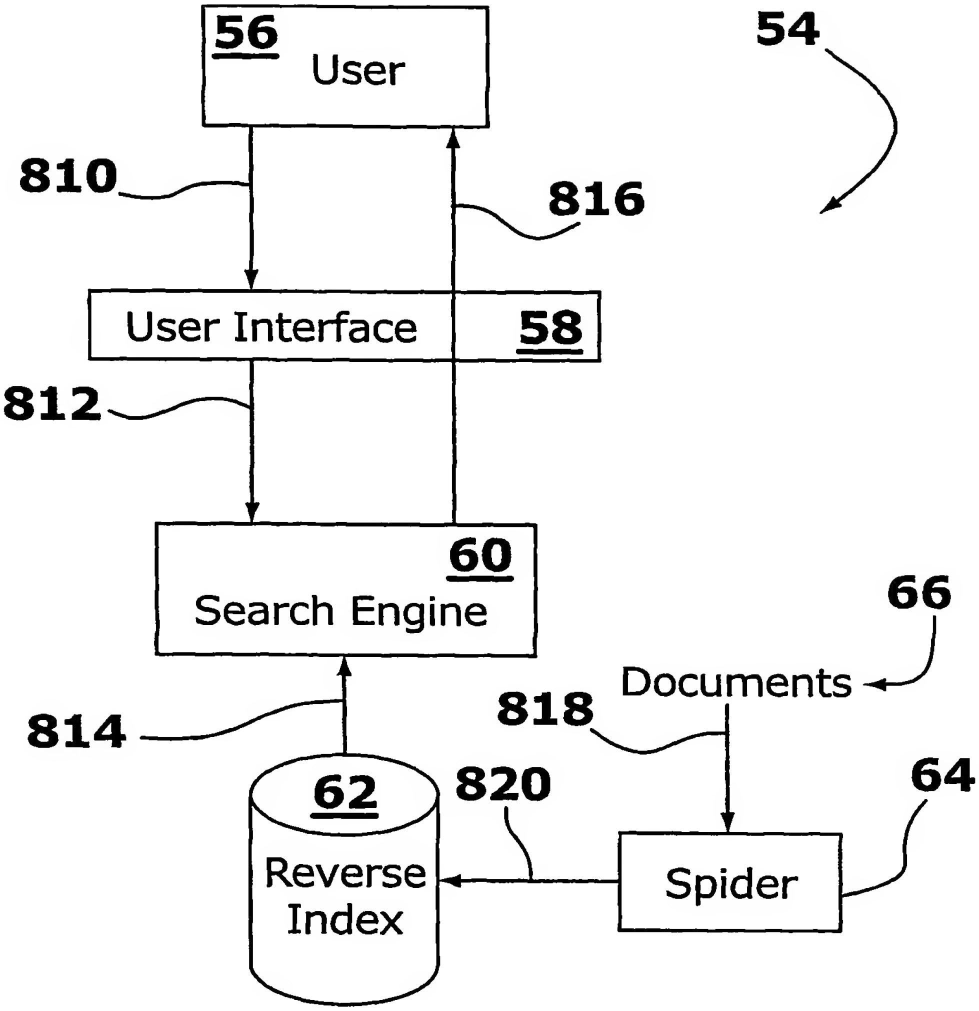


FIG. 8
Prior Art

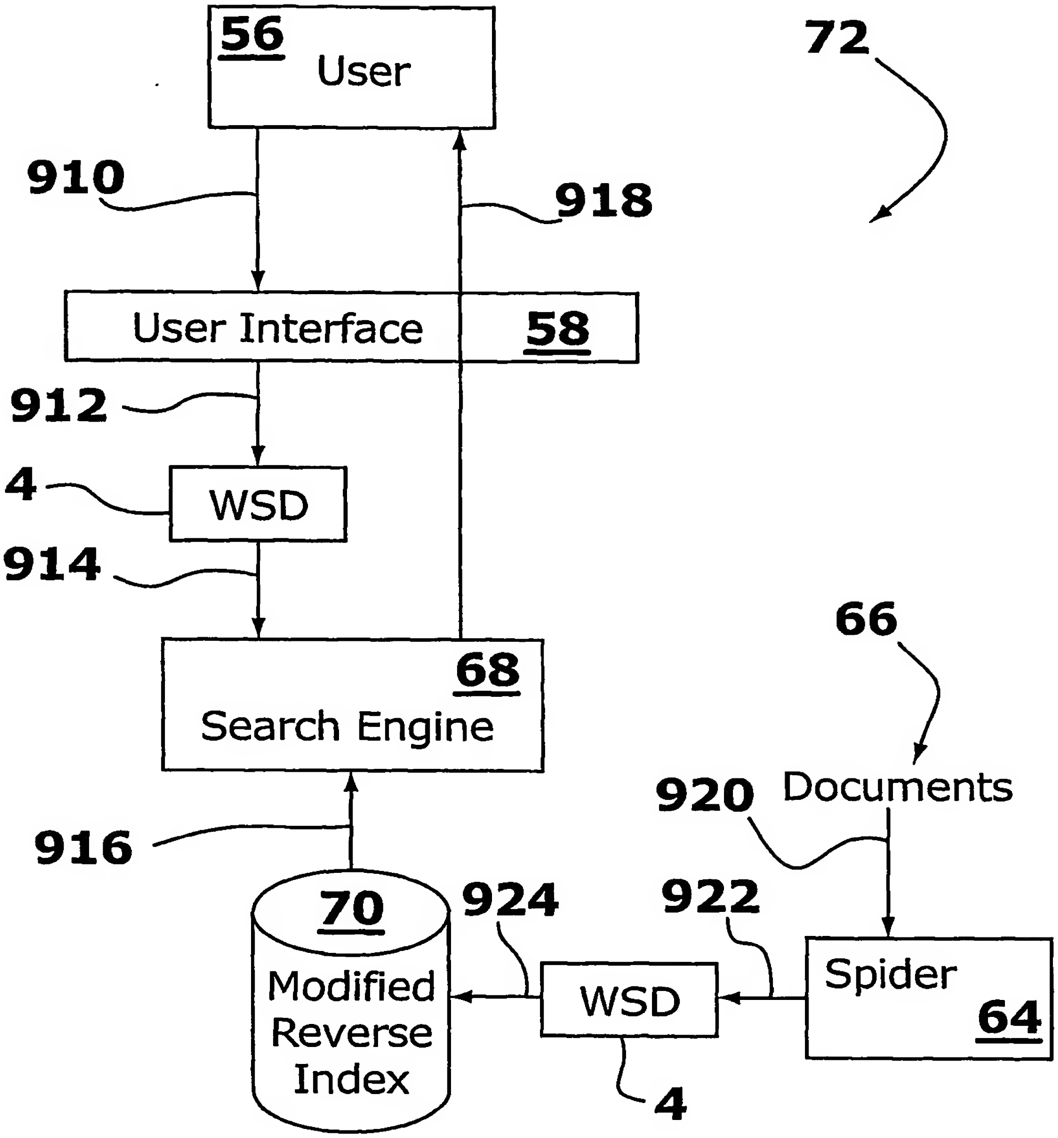

56
User
72
910
918
User Interface
58
912
4
WSD
914
66
68
Search Engine
920
Documents
916
70
Modified
Reverse
Index
924
922
WSD
Spider
64
4

FIG. 9

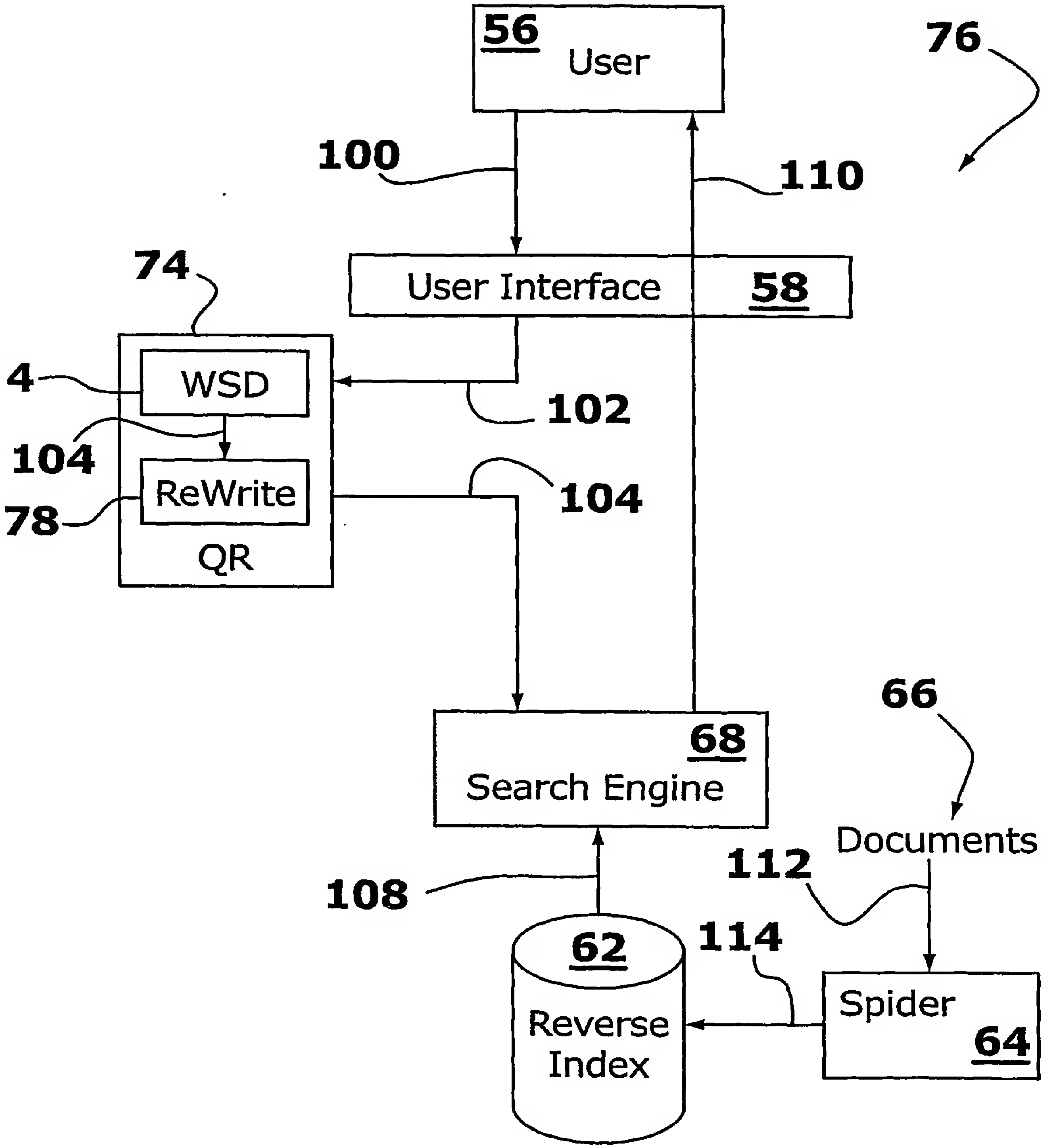

56
User
76
100
110
User Interface
58
74
4
WSD
102
104
ReWrite
104
78
QR
66
68
Search Engine
Documents
112
108
62
114
Spider
Reverse
Index
64

FIG. 10

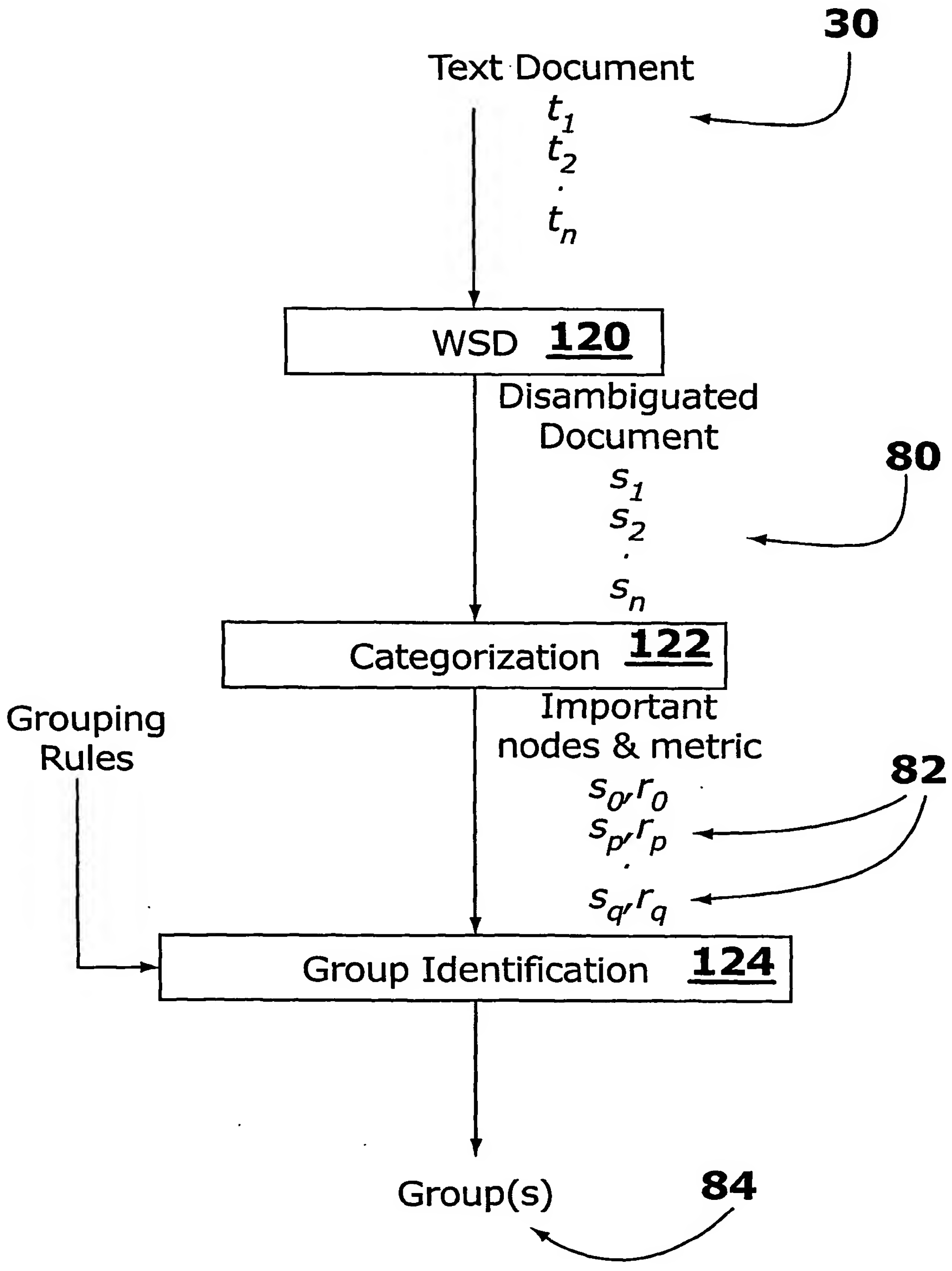

30
Text Document
t_1
t_2
.
t_n
WSD 120
Disambiguated Document
80
s_1
s_2
.
s_n
Categorization 122
Grouping Rules
Important nodes & metric
82
s_0, r_0
s_p, r_p
.
s_q, r_q
Group Identification 124
Group(s)
84

FIG. 11

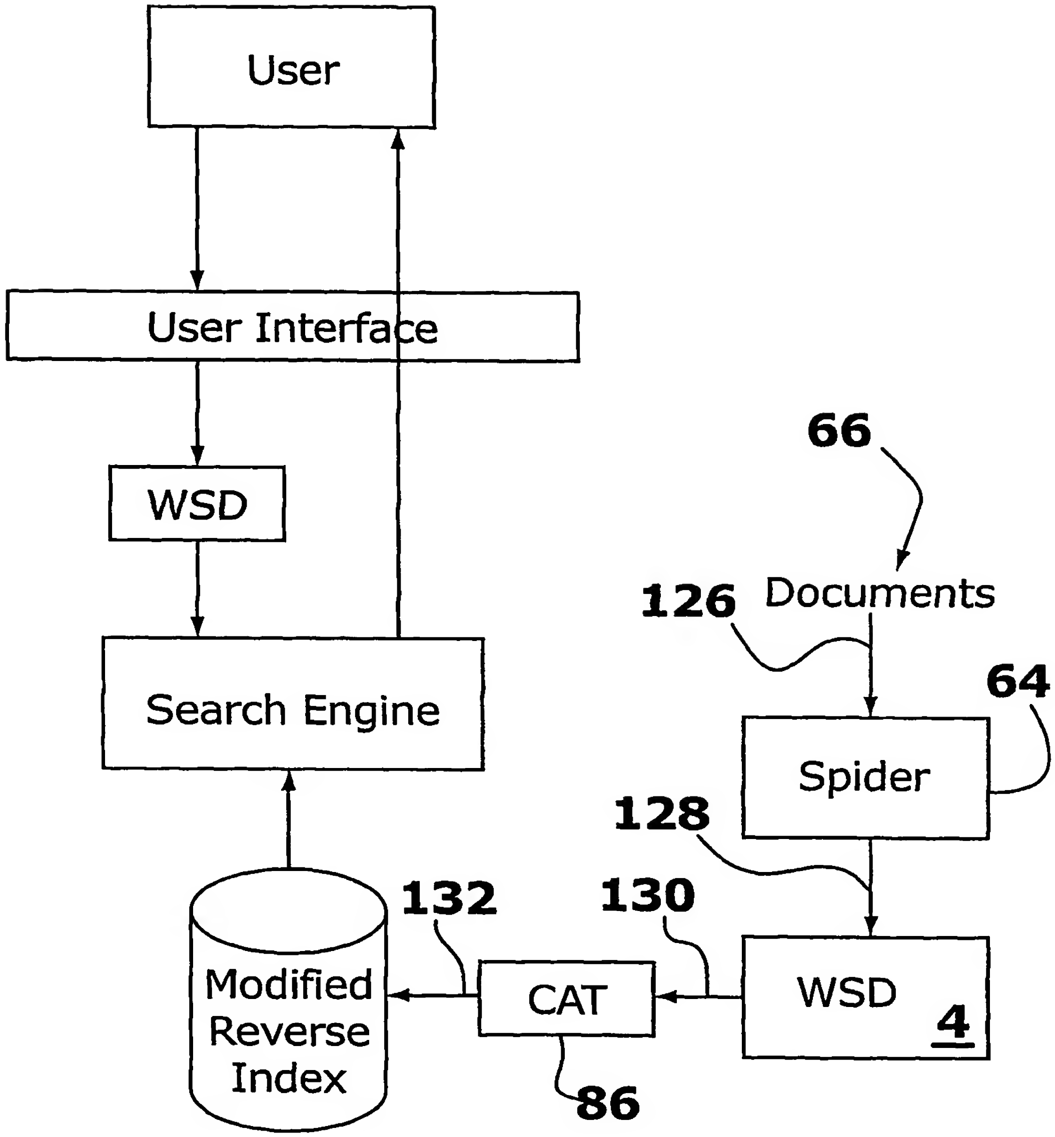
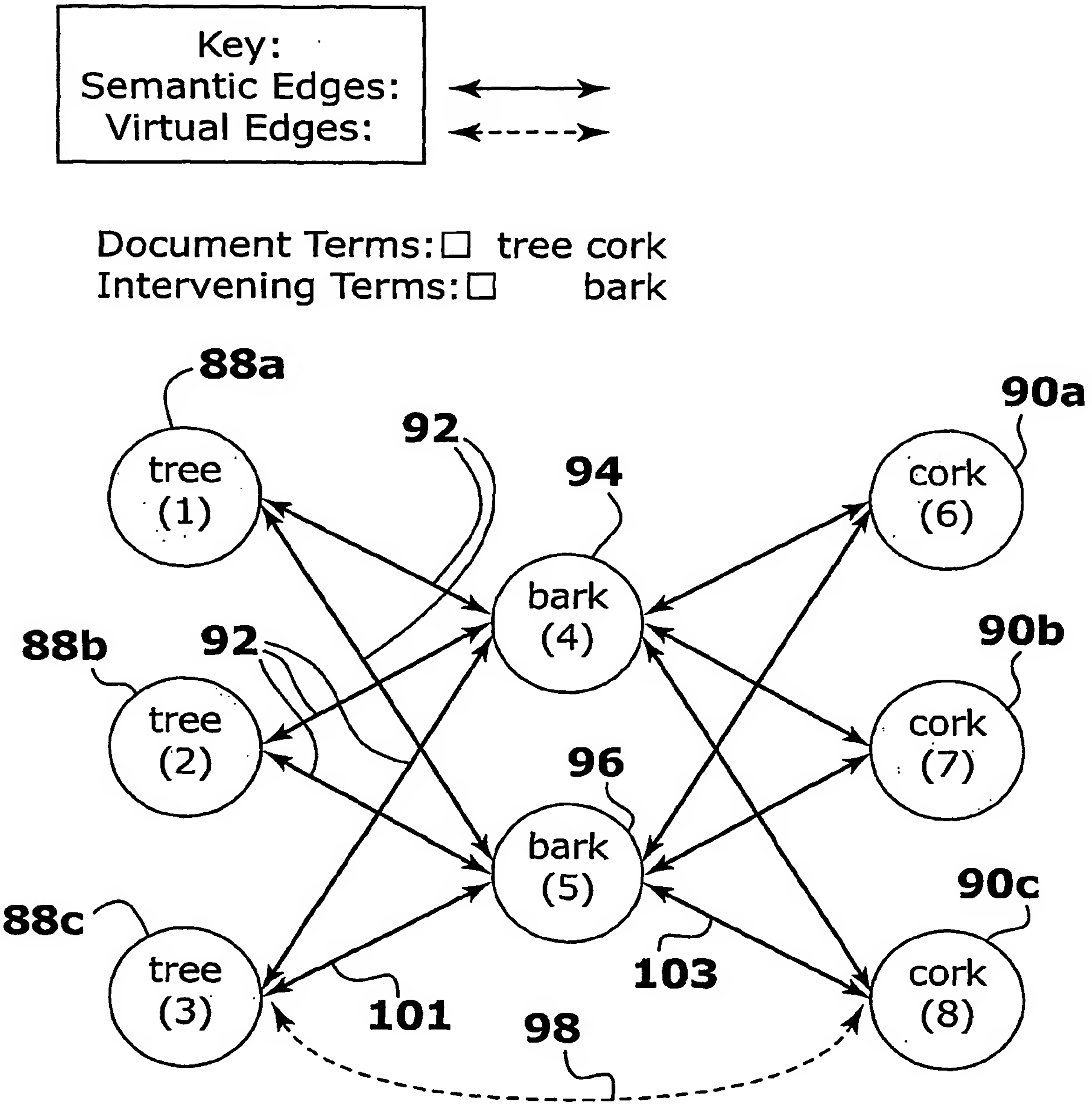

User
User Interface
WSD
Search Engine
Modified Reverse Index
66
Documents
126
64
Spider
128
132
130
CAT
WSD
4
86

FIG. 12



(Note that only one of the virtual edges is shown for clarity.)

FIG. 13

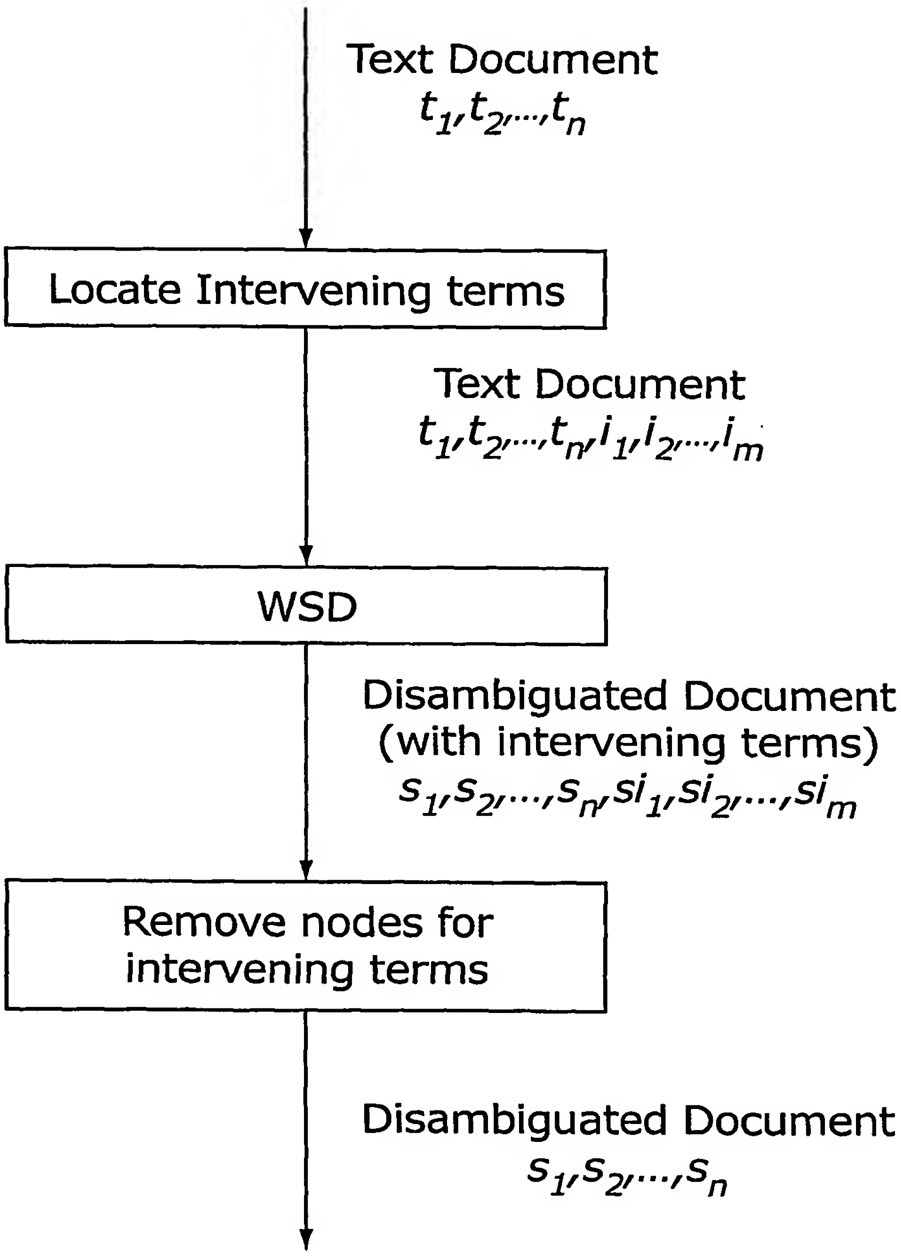

Text Document
$t_1, t_2, \ldots, t_n$
Locate Intervening terms
Text Document
$t_1, t_2, \ldots, t_n, i_1, i_2, \ldots, i_m$
WSD
Disambiguated Document
(with intervening terms)
$s_1, s_2, \ldots, s_n, si_1, si_2, \ldots, si_m$
Remove nodes for intervening terms
Disambiguated Document
$s_1, s_2, \ldots, s_n$

FIG. 14